E.G. GOLSHTEIN

PROBLÈMES PARTICULIERS DE LA PROGRAMMATION LINÉAIRE

E. Golstein, D. Youdine

PROBLÈMES PARTICULIERS DE LA PROGRAMMATION LINÉAIRE

ÉDITIONS MIR

MOSCOU

Table des matières

Préface 7

CHAPITRE PREMIER
RÉSEAUX DE TRANSPORT ET PROBLÈMES DE TRANSPORT 11

§ 1. Réseaux de transport 11
§ 2. Problème de l'itinéraire le moins cher 22
§ 3. Problème du flot maximal 31
§ 4. Formulation en termes de réseaux et formulation matricielle des problèmes de transport 45
§ 5. Méthode des potentiels
§ 6. Méthode hongroise 61
Exercices 76

CHAPITRE 2
PROGRAMMATION LINÉAIRE ET THÉORIE DES JEUX 89

§ 1. Généralités sur la théorie des jeux 89
§ 2. Relation entre jeux matriciels et programmation linéaire 96
§ 3. Méthodes de résolution des jeux 107
§ 4. Méthodes itératives de programmation linéaire 134
Exercices 145

CHAPITRE 3
PROGRAMMATION PARAMÉTRIQUE 148

§ 1. Cas $C = C' + \lambda C''$ 149
§ 2. Cas $B = B' + \mu B''$ 164
§ 3. Cas général 178
§ 4. Applications de la programmation paramétrique 196
§ 5. Sensibilité des solutions des problèmes de programmation linéaire à la variation des données 218
Exercices 234

CHAPITRE 4
PROGRAMMATION PAR BLOCS 238

§ 1. Méthode de décomposition 239
§ 2. Cas particuliers et modifications de la méthode de décomposition 257

§ 3. Méthode duale de discussion des problèmes de programmation par blocs 296
§ 4. Analogue dual de la méthode de décomposition 306
§ 5. Autres méthodes de programmation par blocs fondées sur la minimisation de la fonction $f(\Lambda)$ 310
§ 6. Choix de l'approximation initiale 323
§ 7. Convergence des méthodes de programmation par blocs associées à la minimisation de la fonction $f(\Lambda)$ 327
§ 8. Autre méthode de programmation par blocs 335
§ 9. Application de la méthode de décomposition au problème de transport et à ses variantes 339
§ 10. Problèmes de transport à indices multiples 354
§ 11. Une application de la méthode de décomposition 359
Exercices 362

CHAPITRE 5
PROGRAMMATION LINÉAIRE EN NOMBRES ENTIERS 365

§ 1. Problèmes de programmation linéaire en nombres entiers 366
§ 2. Conditions d'intégralité des ensembles convexes 379
§ 3. Algorithme de programmation en nombres entiers 385
§ 4. Algorithme de programmation partiellement en nombres entiers 407
§ 5. Autres méthodes de programmation en nombres entiers 417
Exercices 428

CHAPITRE 6
PROGRAMMATION STOCHASTIQUE 430

§ 1. Classification des problèmes de programmation stochastique 431
§ 2. Formulation serrée des problèmes de programmation stochastiques. 439
§ 3. Problèmes à contraintes probabilistes 441
§ 4. Problèmes à deux étapes de la programmation stochastique 450
§ 5. Evaluation des solutions approchées 463
§ 6. Programmation markovienne 465
Exercices 468

CHAPITRE 7
PROGRAMMATION LINÉAIRE PAR MORCEAUX 471

§ 1. Programmation linéaire par morceaux et problèmes linéaires 472
§ 2. Critères d'optimalité des problèmes de programmation linéaire par morceaux 482
§ 3. Principes théoriques de la méthode de résolution du problème I 493
§ 4. Dégénérescence 503
§ 5. Algorithme de programmation linéaire par morceaux 507
§ 6. Exemple 518
§ 7. Généralités sur la construction des algorithmes de programmation linéaire par morceaux 536
Exercices 550

BIBLIOGRAPHIE 552

INDEX ALPHABÉTIQUE 558

Préface

Les applications de la programmation linéaire à l'économie, à la technique et à d'autres domaines ne cessent de s'étendre. Des problèmes pratiques nouveaux nécessitent la formulation et la résolution de nouveaux problèmes théoriques et numériques, d'où la nécessité de perfectionner les manières d'aborder les méthodes de programmation linéaire. Les voies que suit la programmation linéaire sont définies essentiellement par les particularités des problèmes spéciaux et les moyens propres aux ordinateurs modernes.

[87] donne un exposé assez détaillé de la théorie et des méthodes finies de résolution du problème général de la programmation linéaire et d'importants problèmes particuliers.

L'ouvrage proposé à l'attention du lecteur est consacré à l'étude de branches caractéristiques de la programmation linéaire. Des conceptions et des méthodes nouvelles y trouvent leur expression, de même que certains problèmes importants auxquels les cours de programmation linéaire accordent généralement une attention insuffisante.

Le présent ouvrage doit être considéré comme faisant suite à [87]. Dans leur exposé les auteurs se sont adressés au lecteur familier avec notions fondamentales, les résultats qualitatifs et les algorithmes de calcul donnés dans [87].

L'ouvrage compte sept chapitres.

Le premier chapitre traite des généralités et des méthodes d'analyse de la théorie des réseaux de transport. Les formulations en termes de réseaux des problèmes du type « transport » sont assez étoffées et concrètes, et rendent possible dans de nombreux cas l'utilisation de méthodes plus économiques par rapport aux formulations matricielles correspondantes. Les applications des méthodes de réseaux dépassent le cadre des problèmes du type « transport ».

Le chapitre 2 est consacré à la liaison fructueuse de la programmation linéaire et de la théorie des jeux, liaison qui permet d'appliquer les méthodes de l'une de ces disciplines à la résolution des problèmes de l'autre. Les méthodes de résolution des jeux sont associées aux

méthodes itératives de programmation linéaire. On considère également des méthodes itératives spécialement adaptées aux problèmes de programmation linéaire.

Le chapitre 3 a pour objet des corrélations entre les composantes du programme optimal d'un problème de programmation linéaire et la variation des paramètres de la forme linéaire, du vecteur des contraintes ou des vecteurs des conditions. Les méthodes de programmation paramétrique exposées ont des applications théoriques et pratiques variées. Dans certains cas elles permettent d'obtenir des estimations qualitatives des solutions des problèmes, dans d'autres, de simplifier la procédure de calcul.

Les méthodes exposées dans le chapitre 3 constituent la base d'une analyse sommaire des familles de problèmes de programmation linéaire à un paramètre.

Le chapitre 4 est consacré à la programmation par blocs. La nécessité pratique de disposer de méthodes spéciales de résolution des problèmes structurés en blocs est déterminée par la grande taille des programmes extrémaux conditionnels dont l'intérêt réside dans les possibilités de les utiliser pour les applications, ainsi que par le volume limité de la mémoire opérationnelle des calculateurs digitaux.

Les méthodes de programmation par blocs conduisent souvent à des schémas de calcul efficaces pour la résolution des problèmes de grande taille. Ces méthodes permettent également de construire des algorithmes spéciaux de programmation linéaire et d'étendre les méthodes bien connues de l'exploration des modèles particuliers à des classes de problèmes plus larges.

Le chapitre 5 examine les problèmes et les méthodes de programmation en nombres entiers. On donne ce nom à la branche de la programmation linéaire dans laquelle toutes ou une partie des variables du problème ne peuvent prendre que certaines valeurs fixées d'avance. De nombreux problèmes extrémaux linéaires et non linéaires, ayant d'importantes applications, peuvent être énoncés en termes de programmation en nombres entiers.

Le chapitre 6 est une tentative de systématisation de la documentation disparate relative à la programmation stochastique. Dans les modèles de programmation linéaire auxquels se ramènent de nombreux problèmes pratiques de gestion et de planification, l'information sur la valeur de certains paramètres (ou de tous les paramètres) des conditions du problème peut s'avérer incomplète. Les moyens de formaliser la position de tels problèmes et les méthodes de résolution constituent l'objet de la programmation stochastique et du chapitre 6.

Le chapitre 7 qui traite de la programmation linéaire par morceaux se situe à la limite entre programmation linéaire et programmation convexe. Ce chapitre expose les principes théoriques, les métho-

des d'analyse et les algorithmes de calcul pour diverses classes de problèmes extrémaux linéaires par morceaux. Les méthodes de programmation linéaire par morceaux sont des généralisations logiques des méthodes finies de programmation linéaire.

La plupart des chapitres de ce livre constituent un exposé systématique et suffisamment détaillé des problèmes particuliers de la programmation linéaire. Cet ouvrage traite d'une série de nouveaux résultats (méthodes de résolution de certains problèmes de réseau; exploration schématisée du problème général à un paramètre de la programmation linéaire et quelques applications de la programmation paramétrique; nouvelles attitudes, suffisamment générales, devant les problèmes de structure par blocs; théorie, méthodes et algorithmes de résolution de diverses classes de problèmes linéaires par morceaux, etc.).

Nous conservons dans ce livre la terminologie, les notations et les méthodes adoptées dans [87]. Les différents chapitres de ce livre sont très peu liés entre eux, et le lecteur familiarisé avec les principes de la programmation linéaire peut aborder l'étude des « Problèmes particuliers » par n'importe quel chapitre. Les auteurs expriment leur gratitude à V. Bokova pour la présentation du manuscrit et la mise au point de nombreux exemples donnés à titre d'illustration.

Les Auteurs.

Chapitre premier

RÉSEAUX DE TRANSPORT ET PROBLÈMES DE TRANSPORT

Ce chapitre est consacré aux éléments de la théorie des réseaux de transport et aux méthodes de résolution des problèmes extrémaux les plus typiques formulés en termes de réseaux.

Le § 1 donne la définition d'un réseau de transport et des notions qui lui sont associées. Le paragraphe suivant traite de la méthode de résolution du problème de réseau le plus simple, celui du chemin le moins cher. Ensuite (§ 3) vient la description de la procédure de calcul du flot maximal, qui est à la base de la méthode hongroise de résolution du problème de transport. Les trois derniers paragraphes ont trait au problème de transport sur un réseau. Le § 4 donne les différents énoncés de ce problème et explique ses relations avec la formulation matricielle des problèmes de transport.

Pour résoudre le problème de transport sur un réseau on recourt à la méthode des potentiels et à sa modalité (§ 5), ainsi qu'à la méthode hongroise (§ 6).

§ 1. Réseaux de transport

1.1. Dans les représentations matricielles des problèmes de transport T et T_d décrites dans les chapitres 12 et 13 de [87], on suppose qu'il n'y a dans chacun de ces problèmes que des points de fabrication et des points de consommation, les transports des points de consommation vers les points de fabrication étant impossibles. Comme nous l'avons dit au chapitre 13 de [87], les problèmes de transport peuvent être représentés graphiquement.

La figure 1.1 est la représentation graphique d'un problème de transport à deux points de fabrication et trois points de consommation où chaque couple de points (fabrication et consommation) est lié par une communication. La figure 1.2 illustre un problème de transport qui comporte des points de fabrication et de consommation entre lesquels le trafic est impossible. Dans le premier cas aussi bien que dans le deuxième les communications sont matérialisées par des flèches allant des points de fabrication aux points de consomma-

tion, ce qui traduit le fait que dans le sens inverse le transport est impossible. Or, si l'on examine une carte ferroviaire (ou routière), il saute aux yeux que la plupart des points peuvent assurer l'entrée et la sortie de la production.

Il est donc naturel d'étendre l'énoncé du problème de transport de façon qu'il puisse couvrir les modèles de transport dont le système

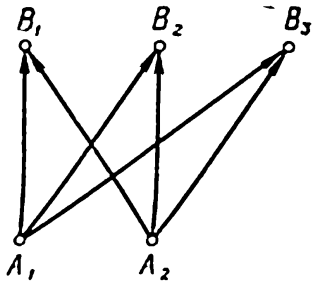

Fig. 1.1

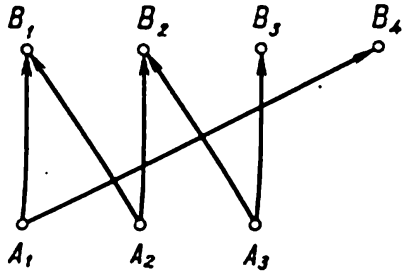

Fig. 1.2

de communications a, par exemple, la structure représentée figure 1.3. Ce système de communications comprend 8 points dont 2 seulement possèdent la propriété supposée dans les modèles du type T et T_d. Le point p_2 ne peut qu'importer la production et le point p_7 ne peut que l'exporter. Les autres points peuvent assurer indépendamment son entrée ou sa sortie.

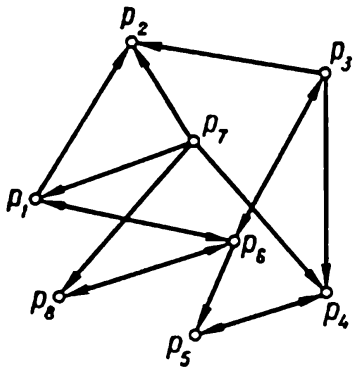

Fig. 1.3

1.2. Voici quelques définitions dont les termes facilitent la formulation des extensions des problèmes de transport qui nous intéressent et certains autres modèles similaires.

La notion de réseau de transport est fondamentale pour ce chapitre.

Soient P un ensemble fini quelconque des éléments p_i, K un ensemble constitué de certains couples ordonnés $k_{ij} = (p_i, p_j)$ (pas nécessairement de tous), dont chacun est composé de différents éléments de l'ensemble P.

Nous appellerons *réseau de transport* (ou simplement réseau) (P, K) la collection de deux ensembles P et K et nous dirons que ce réseau est défini ou engendré par ces ensembles.

Les éléments p_i de l'ensemble P s'appellent *points* du réseau de transport (P, K) et les éléments $k_{ij} = (p_i, p_j)$ de l'ensemble K, *communications* du réseau. Nous conviendrons de dire que la communication k_{ij} relie les points p_i et p_j en commençant en p_i et en se terminant en p_j. Ainsi, donner un réseau de transport signifie indiquer l'ensemble de ses points et les communications qui les relient.

Un réseau de transport quelconque peut être représenté graphiquement. Dans ce but il faut associer à chaque point du réseau

un point du plan, puis relier chaque couple de points (p_i, p_j) qui correspond à la communication du réseau par un segment orienté allant de p_i à p_j. L'orientation du segment est indiquée par une flèche. Si pour certains i et j le réseau comporte à la fois les communications k_{ij} et k_{ji}, dites alors opposées, le segment qui relie p_i et p_j ou bien est affecté de deux flèches dirigées en sens opposés, ou bien ne comporte pas de flèche.

Les figures 1.1 à 1.3 représentent trois réseaux de transport différents. A chaque couple non ordonné de points du réseau de la figure 1.1 correspond une communication unique. Le réseau de la figure 1.3 comporte des points liés par deux communications opposées (par exemple, les points p_4 et p_5). Comme on le voit d'après les figures, les communications relient non pas tous les points, mais seulement certains d'entre eux.

Les réseaux de transport peuvent être également représentés par des matrices.

Supposons que le réseau (P, K) compte N points. Considérons la matrice carrée A d'ordre N dont les éléments a_{ij} se calculent suivant la règle

$$a_{ij} = \begin{cases} 1, \text{ si la communicaction } k_{ij} \in K, \\ 0 \text{ dans le cas contraire.} \end{cases}$$

La matrice A définit univoquement le réseau de transport (P, K) qui l'avait engendrée; elle s'appelle matrice de ce réseau. Ainsi, les matrices A' et A'' des réseaux (fig. 1.2 et 1.3) ont la forme

$$A' = \begin{array}{c} \\ A_1 \\ A_2 \\ A_3 \\ B_1 \\ B_2 \\ B_3 \\ B_4 \end{array} \begin{array}{c} \begin{array}{ccccccc} A_1 & A_2 & A_3 & B_1 & B_2 & B_3 & B_4 \end{array} \\ \left\| \begin{array}{ccccccc} 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array} \right\| \end{array};$$

$$A'' = \begin{array}{c} \\ p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \\ p_6 \\ p_7 \\ p_8 \end{array} \begin{array}{c} \begin{array}{cccccccc} p_1 & p_2 & p_3 & p_4 & p_5 & p_6 & p_7 & p_8 \end{array} \\ \left\| \begin{array}{cccccccc} 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{array} \right\| \end{array}.$$

L'examen d'une ligne de la matrice A' ou A'' révèle les points vers lesquels le trafic peut s'établir à partir du point correspondant à la ligne considérée. Les colonnes des matrices permettent de déterminer les communications qui se terminent aux points choisis. La diagonale principale de la matrice d'un réseau de transport est toujours composée de zéros, car par définition il n'existe pas de communications qui commencent et se terminent au même point.

Une particularité caractéristique de la matrice A' est que ses lignes relatives aux points de consommation et ses colonnes relatives aux points de fabrication ne comportent que des zéros. Cette particularité est évidemment propre à la matrice de tout réseau de transport n'ayant pas de points qui soient simultanément origine et extrémité de communications différentes. Les réseaux de ce type peuvent être définis par des matrices rectangulaires $m \times n$, où $m + n = N$. Les lignes de la matrice correspondent aux origines des communications, et les colonnes, à leurs extrémités. Ainsi, le réseau de transport de la figure 1.2 peut être représenté non seulement par la matrice A', mais encore de façon plus concise en se servant de la matrice

$$A''' = \begin{array}{c} \\ A_1 \\ A_2 \\ A_3 \end{array} \begin{array}{c} \begin{array}{cccc} B_1 & B_2 & B_3 & B_4 \end{array} \\ \left\| \begin{array}{cccc} 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \end{array} \right\| \end{array}.$$

Il convient de noter que dans le calcul manuel la représentation graphique d'un réseau de transport est sans aucun doute préférable aux modes analytiques car elle est plus claire et fait appel à l'intuition.

1.3. Considérons la suite des communications

$$k_{i_1 j_1},\ k_{i_2 j_2},\ \ldots,\ k_{i_s j_s} \tag{1.1}$$

d'un réseau de transport (P, K).

Si chaque couple de communications adjacentes (1.1) compte un point commun alors que les autres couples de communications n'en ont pas, la suite des communications (1.1) s'appelle *itinéraire*.

Nous conviendrons de dire que l'itinéraire (1.1) passe par les points qui sont reliés par ses communications $k_{i_\lambda j_\lambda}$. Chacune des communications extrêmes du chemin (1.1), $k_{i_1 j_1}$ et $k_{i_s j_s}$, compte un point qui ne fait pas partie d'autres communications de l'itinéraire. Ces deux points s'appellent *points extrémaux* de l'itinéraire. Lorsque p_i et p_j sont les points extrémaux d'un itinéraire on dit qu'ils sont reliés par ce chemin.

Si le chemin passe par un point qui n'est pas un point extrémal, ce point fait partie de deux communications du chemin considéré. La définition du chemin entraîne que son image géométrique est une ligne brisée non fermée qui ne passe deux fois par aucun de ses

sommets. La figure 1.4 représente deux chemins du réseau de transport donnés par la figure 1.3. Les suites de communications de la figure 1.5 ne sont pas des chemins puisque chacune d'elles passe par les points qui font partie de plus de deux communications (points p_2 et p_7).

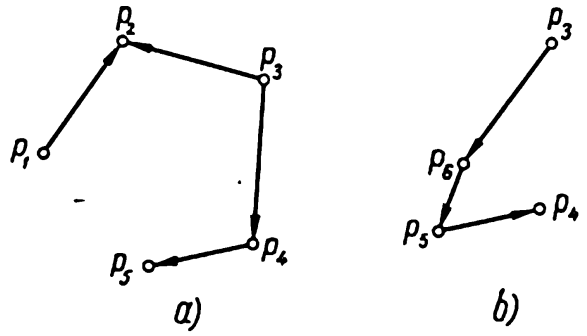

Fig. 1.4

Par définition les points p_i et p_j doivent vérifier la condition $p_i \neq p_j$. Supposons maintenant que $p_i = p_j$. La suite de communications correspondante s'appelle alors *itinéraire fermé*. Il est

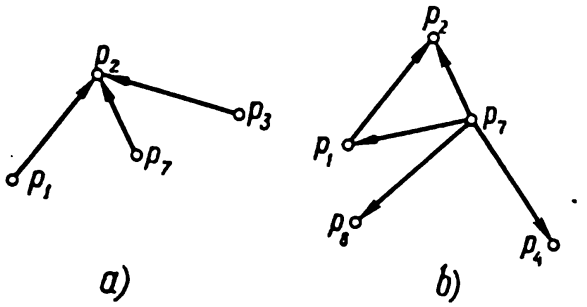

Fig. 1.5

clair que tout point par lequel passe un itinéraire fermé appartient à deux communications adjacentes de l'itinéraire (et seulement à ces communications).

Des exemples d'itinéraires fermés sont donnés par la figure 1.6. Si un point quelconque d'un itinéraire l, autre que point extrémal, marque l'origine d'une communication et la fin d'une autre, l'itinéraire l s'appelle alors *itinéraire orienté*.

Le point extrémal d'un itinéraire orienté où commence (finit) l'une de ses communications est dit *origine* (*extrémité*).

On définit de façon analogue un itinéraire *orienté fermé*. Chaque point d'un tel itinéraire constitue l'extrémité d'une de ses communications et l'origine d'une autre.

Un exemple d'itinéraire orienté est représenté sur la figure 1.4, *b*. Un itinéraire orienté fermé est donné par la figure 1.6, *a*.

Un réseau de transport engendré par certaines parties des ensembles P et K s'appelle sous-réseau du réseau de transport (P, K).

Un réseau de transport est dit *connexe* si tout couple de points de ce réseau peut être relié par un de ses chemins.

Un sous-réseau connexe d'un réseau de transport (P, K) est *maximal* s'il contient avec un point quelconque p_i le point $p_j \in P$

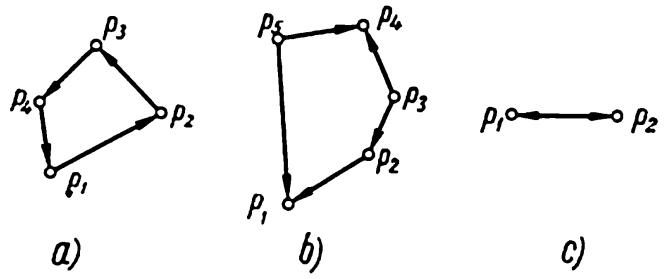

Fig. 1.6

relié à p_i par une communication $k_{ij} \in K$ ainsi que toutes les communications de l'ensemble K qui joignent les points de ce sous-réseau.

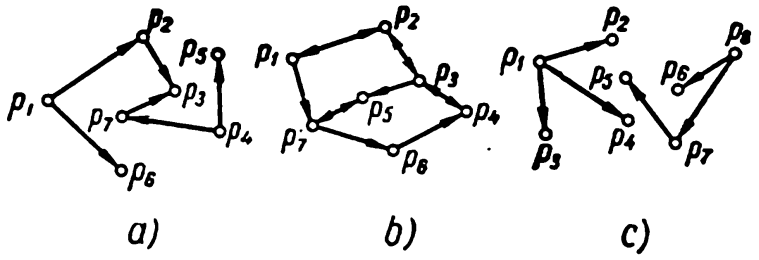

Fig. 1.7

On vérifie facilement qu'un réseau de transport arbitraire se divise d'une façon et d'une seule en un nombre fini de sous-réseaux connexes maximaux (cf. exercice 1).

La figure 1.7. donne des exemples de réseaux connexes et non connexes.

Si chaque couple de points d'un réseau de transport peut être associé par un chemin orienté, on dit que le réseau est *connexe orienté*. Il est clair que tout réseau connexe orienté est connexe. La réciproque est fausse : il existe des réseaux connexes qui ne sont pas connexes orientés. Un exemple en est fourni par le réseau représenté fig. 1.7, *a*. Le réseau de la fig. 1.7, *b* est connexe orienté.

Il convient de noter que le concept de réseau de transport ainsi introduit est un cas particulier de celui de graphe, largement utilisé dans diverses applications. Dans la mesure où nous limitons notre intérêt aux applications des graphes aux problèmes de transport, nous avons cru nécessaire d'adapter la terminologie imposée à la nature même de ces applications. Dans ce qui suit le terme de graphe

ne s'emploie pas, et tous les résultats indispensables de la théorie des graphes sont énoncés et démontrés en fonction des réseaux de transport.

1.4. Nous aurons ci-dessous à examiner plusieurs fonctions définies soit sur l'ensemble des points d'un réseau, soit sur l'ensemble de ses communications.

Soit $f(p_i)$ une fonction réelle des points $p_i \in P$; $g(k_{ij})$, $k_{ij} \in K$, une fonction réelle définie sur l'ensemble des communications K.

Introduisons les notations suivantes pour les sommes de valeurs de ces fonctions, notations qu'il est commode d'utiliser dans une série de problèmes de la théorie des réseaux de transport:

$f(P') = \sum_{p_i \in P'} f(p_i)$ est la somme des valeurs de la fonction f sur un ensemble de points $P' \subset P$;

$g(P', P'') = \sum_{\substack{p_i \in P' \\ p_j \in P''}} g(k_{ij})$, la somme des valeurs de la fonction g sur l'ensemble de toutes les communications commençant en un des points de l'ensemble $P' \subset P$ et finissant en les points de l'ensemble $P'' \subset P$.

Voici les propriétés évidentes des notations adoptées.

Si P', P'', P''' sont des parties de l'ensemble P, si de plus P'' et P''' n'ont pas de points communs ($P'' \cap P''' = 0$),

$$f(P'' \cup P''') = f(P'') + f(P'''), \tag{1.2}$$

$$g(P', P'' \cup P''') = g(P', P'') + g(P', P'''), \tag{1.3}$$

$$g(P'' \cup P''', P') = g(P'', P') + g(P''', P'). \tag{1.4}$$

Rappelons que $P'' \cup P'''$ ($P'' \cap P'''$) est l'ensemble des points appartenant à l'ensemble P'' ou à l'ensemble P''' (appartenant à la fois à P'' et à P''').

Enumérons quelques fonctions qui figurent généralement dans les problèmes relatifs aux réseaux de transport.

1. La fonction de fabrication et de consommation $q(p_i)$ est donnée sur l'ensemble P de tous les points du réseau (P, K). Si $q_i = q(p_i) > 0$, p_i est dit *point de fabrication*; si $q_i = q(p_i) < 0$, p_i est dit *point de consommation*. p_i pour lequel $q_i = q(p_i) = 0$ s'appelle *point de transit*.

Le sens des fonctions $q(p_i)$, $p_i \in P$ est évident. Le nombre $q(p_i)$ est égal au volume de la fabrication d'un produit homogène en p_i lorsque $q(p_i) > 0$; il est égal au volume de la consommation du produit homogène en p_i, pris avec le signe moins, lorsque $q(p_i) < 0$. La valeur nulle de $q(p_i)$ indique qu'en p_i il n'y a ni fabrication ni consommation du produit. Les ensembles des points de fabrication, de consommation et de transit seront parfois désignés par P_+, P_- et P_0 respectivement.

2. La fonction des capacités des communications $d(k_{ij}) \geqslant 0$ est définie sur l'ensemble K des communications du réseau de tran-

sport (P, K). La valeur de la fonction des capacités de la communication k_{ij} est égale à la quantité limite de produit susceptible d'être transportée par cette communication. Le nombre $d_{ij} = d(k_{ij})$ s'appelle capacité de la communication k_{ij}. Si la communication donnée est capable d'assurer le transport d'une quantité quelconque de produit, sa capacité est considérée infinie (∞). Si le réseau (P, K) ne compte pas de communication allant du point p_λ au point p_μ, on peut poser par convention $k_{\lambda\mu} \in K$, mais $d_{\lambda\mu} = d(k_{\lambda\mu}) = 0$.

3. Considérons le réseau (P, K) sur l'ensemble des communications duquel est définie la fonction des capacités $d(k_{ij})$, $k_{ij} \in K$. Choisissons deux points du réseau p_1 et p_N et appelons p_1 entrée et p_N sortie. Soient E_i' l'ensemble des communications du réseau partant du point p_i et E_i'', l'ensemble des communications du réseau qui finissent au point p_i.

Une fonction $x_{ij} = x(k_{ij})$, définie sur l'ensemble K, satisfaisant aux relations

$$\sum_{k_{ij} \in E_i'} x(k_{ij}) - \sum_{k_{ji} \in E_i''} x(k_{ji}) = 0 \text{ pour tout } i \neq 1, N, \tag{1.5}$$

$$x_1 = \sum_{k_{1j} \in E_1'} x(k_{1j}) - \sum_{k_{j1} \in E_1''} x(k_{j1}) \geqslant 0, \tag{1.6}$$

$$0 \leqslant x(k_{ij}) \leqslant d(k_{ij}), \quad k_{ij} \in K, \tag{1.7}$$

s'appelle *flot du réseau* (P, K) compatible avec la fonction $d(k_{ij})$, $k_{ij} \in K$ et dirigé de p_1 vers p_N. En additionnant les relations (1.5), (1.6) et l'identité

$$x_N = \sum_{k_{Nj} \in E_N'} x(k_{Nj}) - \sum_{k_{jN} \in E_N''} x(k_{jN}),$$

on obtient

$$x_1 + x_N = \sum_{i=1}^{N} \Big(\sum_{k_{ij} \in E_i'} x(k_{ij}) - \sum_{k_{ji} \in E_i''} x(k_{ji}) \Big). \tag{1.8}$$

Toute communication part d'un point du réseau et se termine en un autre. C'est pourquoi la grandeur $x(k_{ij})$, i et j étant quelconques, fait deux fois partie du deuxième membre de (1.8), d'abord avec le signe plus puis avec le signe moins,

$$x_1 + x_N = 0. \tag{1.9}$$

Le nombre $x_1 = -x_N \geqslant 0$ s'appelle *valeur du flot* $x(k_{ij})$, $k_{ij} \in K$. Le sens des termes ainsi introduits peut être interprété de la façon suivante. Imaginons que le réseau (P, K) soit composé de canaux à entrée unique p_1 et à sortie unique p_N. Supposons que le canal correspondant à la communication k_{ij} soit capable de laisser passer au plus d_{ij} unités de liquide par unité de temps.

La fonction $x(k_{ij})$ qui satisfait aux conditions (1.5)-(1.7) définit alors les vitesses d'écoulement du liquide dans chaque canal; x_1 est la vitesse d'amenée du liquide à l'entrée p_1, $-x_N$ est la vitesse d'évacuation du liquide par la sortie p_N (dans les conditions d'un régime stationnaire). A la lumière de l'interprétation donnée l'égalité (1.9) est évidente: la quantité de liquide fournie en une unité de temps par l'entrée est égale à celle qui s'échappe pendant le même temps par la sortie.

La fonction introduite $x(k_{ij})$, $k_{ij} \in K$ est non négative, ce qui autorise l'appellation de flot *arithmétique* qu'on lui donne parfois.

Pour mettre en évidence les différentes propriétés d'un flot il est souvent plus commode d'utiliser la notion de flot algébrique, dont les composantes ne sont pas limitées par la condition de non-négativité. Soit $x(k_{ij})$ le flot arithmétique. Posons pour deux points quelconques p_i et p_j, reliés par une communication du réseau,

$$x'_{ij} = x'(k_{ij}) = x(k_{ij}) - x(k_{ji}). \tag{1.10}$$

Si l'une des communications opposées k_{ij} et k_{ji} qui interviennent dans l'égalité (1.10) n'appartient pas à K, la valeur du flot dans cette communication est alors supposée nulle.

Etant donné que $x(k_{ij}) \leqslant d_{ij}$, $x(k_{ji}) \geqslant 0$ et $x(k_{ij}) = 0$ avec $k_{ij} \notin K$, on a

$$x'(k_{ij}) \leqslant d_{ij}, \text{ si } k_{ij} \in K; \tag{1.11}$$

$$x'(k_{ij}) \leqslant 0, \text{ si } k_{ij} \notin K, \text{ mais } k_{ji} \in K. \tag{1.12}$$

Etendons l'ensemble K en y introduisant les communications $k_{ji} \notin K$ sous la condition que $k_{ij} \in K$. Désignons le nouvel ensemble de communications par K' et appelons-le extension de l'ensemble K. Les valeurs de la fonction $d(k_{ij})$ pour les communications nouvellement introduites sont supposées nulles.

Les contraintes (1.11), (1.12) peuvent s'écrire maintenant sous la forme unique

$$x'(k_{ij}) \leqslant d(k_{ij}) \text{ avec } k_{ij} \in K'. \tag{1.13}$$

On a évidemment

$$x'(k_{ij}) = -x'(k_{ji}). \tag{1.14}$$

(1.14) permet d'écrire les conditions (1.5), (1.6) sous une forme plus condensée

$$\sum_{k_{ij} \in E_i} x'(k_{ij}) = \begin{cases} 0 \text{ pour } & i \neq 1, N; \\ x_1 \geqslant 0, & i = 1; \\ x_N \leqslant 0, & i = N, \end{cases} \tag{1.15}$$

où $E_i = E'_i + E''_i$.

La dernière condition du système (1.15) donnée pour la symétrie est une conséquence des conditions précédentes. La fonction $x'(k_{ij})$

définie sur l'ensemble des communications K' et vérifiant les conditions (1.13)-(1.15) s'appelle flot *algébrique* compatible avec la fonction $d(k_{ij})$, $k_{ij} \in K'$.

Comme nous venons de le montrer, à chaque flot arithmétique correspond un flot algébrique compatible avec la même fonction des capacités. Il est facile de vérifier que la réciproque est vraie également : un flot algébrique quelconque engendre un flot arithmétique, les deux flots étant compatibles avec la même fonction des capacités. En effet, soit $x'(k_{ij})$, $k_{ij} \in K'$, un flot algébrique compatible avec la fonction $d(k_{ij})$, $k_{ij} \in K'$.

Posons pour $k_{ij} \in K$

$$x(k_{ij}) = \begin{cases} x'(k_{ij}) \text{ avec } x'(k_{ij}) \geqslant 0, \\ 0 \text{ avec } x'(k_{ij}) < 0. \end{cases} \tag{1.16}$$

(1.13) conduit à la condition (1.7). Les relations (1.14), (1 15) amènent les conditions (1.5), (1.6). Par conséquent, la fonction $x(k_{ij})$ est effectivement un flot arithmétique compatible avec la fonction $d(k_{ij})$, $k_{ij} \in K$.

Si certains points du réseau sont raccordés par deux communications opposées, un flot algébrique peut évidemment engendrer une infinité de flots arithmétiques. Cependant, sous l'hypothèse supplémentaire de la valeur nulle du flot arithmétique dans l'une des communications opposées [la relation (1.16) justifie cet impératif], la correspondance entre les flots algébrique et arithmétique devient biunivoque.

4. Supposons qu'une fonction de fabrication et de consommation $q(p_i)$ soit donnée sur l'ensemble P des points du réseau (P, K) et qu'une fonction des capacités $d(k_{ij})$ soit définie sur l'ensemble K des communications du réseau.

Une fonction $x(k_{ij})$, définie sur l'ensemble K des communications, qui vérifie les relations

$$\sum_{k_{ij} \in E_i'} x(k_{ij}) - \sum_{k_{ji} \in E_i''} x(k_{ji}) = q(p_i) \text{ pour tous les } p_i \in P, \tag{1.17}$$

$$0 \leqslant x(k_{ij}) \leqslant d(k_{ij}), \tag{1.18}$$

s'appelle *programme de transport* compatible avec la fonction de fabrication et de consommation $q(p_i)$, $p_i \in P$ et la fonction des capacités des communications $d(k_{ij})$, $k_{ij} \in K$. En additionnant les égalités (1.17) et en retenant que chaque transport $x(k_{ij})$ entre deux fois dans le premier membre de la relation sommée avec des signes différents,

$$\sum_{p_i \in P} q(p_i) = q(P) = 0. \tag{1.19}$$

L'égalité (1.19), dite condition de bilan, signifie que le volume total de fabrication des points du réseau est égal au volume total

de consommation de ces points. D'après ce qui précède, la relation (1.19) est une condition nécessaire d'existence d'un programme compatible avec la fonction de fabrication et de consommation $q(p_i)$, $p_i \in P$.

Suivant les conditions (1.17) et (1.18), un programme de transport $x(k_{ij})$ compatible avec $q(p_i)$, $p_i \in P$, et $d(k_{ij})$, $k_{ij} \in K$, est une séquence de transports par les communications du réseau qui permet de satisfaire aux demandes de tous les points de consommation grâce aux ressources disponibles aux points de fabrication du réseau donné sans dépasser les limites des capacités de ses communications. Comme dans le cas des flots, les systèmes de transports peuvent être donnés soit sous la forme de programmes arithmétiques soit sous celle de programmes algébriques.

La fonction qui vient d'être introduite est un programme *arithmétique* de transport, car $x(k_{ij}) \geqslant 0$.

Il est parfois plus commode d'opérer avec des programmes algébriques de transport.

Une fonction $x'(k_{ij})$ définie sur l'ensemble des communications K' et vérifiant les conditions

$$\sum_{k_{ij} \in E_i} x'(k_{ij}) = q(p_i), \quad p_i \in P, \tag{1.20}$$

$$x'(k_{ij}) \leqslant d(k_{ij}), \; k_{ij} \in K', \tag{1.21}$$

$$x'(k_{ij}) = -x'(k_{ji}), \; k_{ij} \in K', \tag{1.22}$$

s'appelle programme de transport *algébrique* compatible avec les fonctions $q(p_i)$, $p_i \in P$ et $d(k_{ij})$, $k_{ij} \in K'$. Les notations E_i et K' ont ici le même sens que dans la définition du flot algébrique, et de plus, comme auparavant, $d(k_{ij}) = 0$ si $k_{ij} \notin K$.

De même que dans le cas des flots, on découvre aisément que les formules (1.10) et (1.16) établissent une correspondance entre les programmes arithmétiques et algébriques, qui ne sont que des écritures différentes d'un même système de transports.

5. Dans ce qui suit nous aurons à comparer des programmes de transport du point de vue du coût de leur réalisation. C'est à cette fin que sert la fonction des coûts de transport $c(k_{ij}) \geqslant 0$, définie sur l'ensemble K des communications du réseau (P, K). La valeur de la fonction $c(k_{ij})$, $k_{ij} \in K$, dans la communication k_{ij} est égale au coût de transport d'une unité de produit d'un même genre le long de k_{ij}. Le nombre $c_{ij} = c(k_{ij})$ est appelé parfois coût de transport de la communication k_{ij}. Les fonctions $d(k_{ij})$, $x(k_{ij})$ et $c(k_{ij})$, définies sur des ensembles de communications, peuvent être données soit directement sur un réseau de transport, soit sous forme de matrices carrées d'ordre N [N est égal au nombre de points du réseau (P, K)].

Si $k_{ij} \in K$ (ou $k_{ij} \in K'$ dans le cas de flots et de programmes de transport algébriques), la valeur de la fonction correspondante sur la communication k_{ij} est figurée à l'intersection de la i-ème ligne et de la j-ième colonne de la matrice.

Dans le cas contraire, on porte dans cette case un zéro pour les fonctions $d(k_{ij})$ et $x(k_{ij})$ et ∞ pour la fonction $c(k_{ij})$.

Ainsi les fonctions introduites des communications du réseau peuvent être considérées comme définies pour chaque communication, indépendamment de son appartenance à l'ensemble K. Par ailleurs, l'extension de la fonction aux communications $k_{ij} \notin K$ s'effectue par le procédé décrit dans ce qui précède. Les fonctions définies sur l'ensemble des communications et les matrices qui leur sont associées sont notées parfois par des majuscules correspondantes (D, X, C).

§ 2. Problème de l'itinéraire le moins cher

2.1. Considérons un réseau de transport quelconque (P, K), sur l'ensemble des communications duquel est définie une fonction du coût de transport non négative $c(k_{ij}) = c_{ij}$, $k_{ij} \in K$. Associons à chaque itinéraire orienté

$$l = (k_{i_1 i_2}, k_{i_2 i_3}, \ldots, k_{i_{s-2}\, i_{s-1}}, k_{i_{s-1} i_s})$$

du réseau (P, K) le coût de transport suivant cet itinéraire d'une unité de produit homogène, égal à

$$c(l) = c_{i_1 i_2} + c_{i_2 i_3} + \ldots + c_{i_{s-2}\, i_{s-1}} + c_{i_{s-1}\, i_s}.$$

Appelons le nombre $c(l)$ *valeur du coût de transport* de l'itinéraire l. Le problème de l'itinéraire le moins cher consiste à rechercher l'itinéraire l^*, commençant et finissant aux points fixés du réseau et qui correspond à la valeur minimale du coût de transport par rapport aux autres itinéraires orientés reliant les points considérés dans la direction donnée. L'itinéraire l^* est dit *optimal*.

L'interprétation du problème ainsi énoncé est déterminée par le sens physique de la fonction $c(k_{ij})$, $k_{ij} \in K$.

Si $c(k_{ij})$ est la longueur de la communication k_{ij}, le problème de l'itinéraire le moins cher se transforme en problème de la recherche de l'itinéraire le plus court entre deux points connu sous le nom de *problème du chemin le plus court*. Si $c(k_{ij})$ est le temps nécessaire pour passer du point p_i en p_j, le problème consiste à déterminer l'itinéraire suivant lequel on peut aller en un temps minimal d'un point fixé en un autre point.

Le transport n'est pas l'unique domaine d'application de l'itinéraire le moins cher. Une des tâches des plus importantes de la méthode moderne de planification et de gestion des systèmes (méthode PERT) consiste à déterminer les chemins critiques, c'est-à-dire la

succession de travaux limitant l'exécution d'une commande en fonction d'un certain facteur (temps, frais, telle ou telle ressource). Le problème de la recherche du chemin critique se ramène au choix de l'itinéraire le plus cher d'un réseau de transport parmi ceux qui relient deux de ses points. Ce dernier problème, défini par la fonction C, est équivalent à celui de l'itinéraire le moins cher défini par la fonction $-C$ *).

Le problème de l'itinéraire le moins cher présente un intérêt non seulement de par lui-même en raison des positions des problèmes qui viennent d'être passées en revue; son importance est due encore au fait qu'il entre comme partie intégrante dans plusieurs problèmes de réseau plus complexes. On trouve dans la littérature un grand nombre de méthodes de résolution du problème de l'itinéraire le moins cher. Un aperçu de certaines d'entre elles est donné par [63].

Dans ce paragraphe nous exposons deux algorithmes. Le premier permet de relier un point fixé du réseau à tous les autres points par des itinéraires optimaux; le deuxième est conçu pour la construction simultanée des itinéraires optimaux entre deux points quelconques du réseau.

2.2 Soit p_1 un point quelconque du réseau (P, K). Trouver pour tout point $p_s \in P$ un itinéraire optimal allant de p_1 à p_s. La procédure de résolution de ce problème, décrite ci-dessous, a été proposée par G. Y. Minty [57]. Elle se compose d'un nombre fini d'étapes élémentaires dont chacune consiste à étendre l'ensemble des points du réseau marqués et à dégager certaines de ses communications. On commence en indexant le point p_1, auquel on associe le nombre 0. Supposons que nous ayons déjà effectué plusieurs étapes de l'algorithme qui ont abouti au marquage de certains points du réseau, chacun de ces derniers étant associé à un nombre h_λ, où λ est le numéro du point correspondant. L'étape suivante de l'algorithme consiste à considérer tous les points du réseau non marqués encore qui constituent les extrémités des communications commençant en l'un des points du réseau déjà marqués. Calculons pour chacune de ces communications $k_{\lambda\mu}$ la somme du nombre h_λ associée au point p_λ déjà marqué et les valeurs de son coût de transport $c\,(k_{\lambda\mu})$. Dégageons ensuite celles des communications examinées pour lesquelles la valeur de cette somme est minimale. Dans ces conditions il faut tenir compte de ce que des quelques communications à dégager et qui se terminent en un même point une seule est retenue (peu importe laquelle). On marque les extrémités des communications dégagées en associant à chacune d'entre elles la valeur minimale de la somme. L'étape

*) Bien que les méthodes de résolution du problème de l'itinéraire le moins cher soient adaptées aux fonctions non négatives du coût de transport, les particularités des réseaux établis d'après le système PERT (exempts d'itinéraires fermés orientés) permettent de ne pas tenir compte de cette hypothèse (cf. § 2.5).

s'achève sur cette opération. La procédure se poursuit jusqu'au moment où l'extension ultérieure de l'ensemble des points marqués devient impossible. Par construction, chaque point marqué (sauf p_1) constitue l'extrémité d'une seule communication dégagée. C'est pourquoi la circulation à partir d'un point marqué quelconque p_s sur les communications dégagées (dans la direction opposée à leur orientation) s'effectue de façon univoque. L'origine d'une communication dégagée est marquée avant son extrémité. Par conséquent, la suite des communications dégagées, formée au cours de la circulation à partir du point p_s, constitue l'itinéraire $l^*_{p_1p_s}$ avec l'origine en p_1 et l'extrémité en p_s. Nous allons montrer que l'itinéraire $l^*_{p_1p_s}$ est optimal, c'est-à-dire associé à la plus petite valeur du coût de transport par rapport aux autres itinéraires allant de p_1 à p_s. De plus, la valeur du coût de transport de l'itinéraire $l^*_{p_1p_s}$ est égale à h_s, nombre qui correspond au point p_s.

Servons-nous d'une récurrence sur le numéro t de l'étape à laquelle a été marqué le point p_s. Si $t = 0$, c'est-à-dire si $p_s = p_1$, la proposition est évidente. Supposons qu'elle soit vraie pour tous les points marqués au terme de r étapes au plus. Montrons que pour $t = r + 1$ elle a lieu également. Considérons un itinéraire quelconque $l_{p_1p_s}$ entre p_1 et p_s. Introduisons l'ensemble P_r des points marqués à la suite de r étapes. Par condition, $p_s \in P_{r+1}$, $p_s \notin P_r$. Il est évident que l'itinéraire $l_{p_1p_s}$ doit contenir nécessairement une communication $k_{\alpha\beta}$ telle que $p_\alpha \in P_r$ et $p_\beta \notin P_r$.

D'après l'algorithme

$$h_\alpha + c_{\alpha\beta} \geqslant h_{s'} + c_{s's}, \tag{2.1}$$

où s' est l'origine de la communication dégagée qui se termine en p_s.

D'après l'hypothèse de récurrence, h_α est la valeur du coût de transport de l'itinéraire optimal entre p_1 et p_α. On obtient donc, compte tenu de la non-négativité des quantités c_{ij},

$$c(l_{p_1p_s}) \geqslant h_\alpha + c_{\alpha\beta}. \tag{2.2}$$

Pour la même raison la valeur du coût de transport qui correspond à la partie de l'itinéraire $l^*_{p_1p_s}$ entre p_1 et $p_{s'} \in P_r$ est $h_{s'}$; par conséquent

$$c(l^*_{p_1p_s}) = h_{s'} + c_{s's}. \tag{2.3}$$

La comparaison de (2.1), (2.2) et (2.3) conduit aux relations recherchées

$$h_s = h_{s'} + c_{s's} = c(l^*_{p_1p_s}) \leqslant c(l_{p_1p_s}),$$

qui vérifient notre proposition pour $t = r + 1$. La proposition est ainsi démontrée pour une valeur quelconque du paramètre t.

L'absence d'un point p_λ parmi les points marqués traduit l'impossibilité de former un itinéraire orienté du réseau commençant

en p_1 et finissant en p_λ. En effet, soit P_∞ l'ensemble complet des points marqués. L'existence de l'itinéraire $l_{p_1 p_\lambda}$ entraîne celle d'une communication qui commence en un point de l'ensemble P_∞ et finit en un point non marqué. Mais alors P_∞ n'est pas un ensemble complet des points marqués. Cette réduction à l'absurde justifie notre proposition.

En appliquant l'algorithme décrit ci-dessus à des problèmes concrets il est utile de retenir les observations suivantes.

1. Il peut arriver que les itinéraires optimaux issus du point p_1 doivent relier non pas tous les points du réseau, mais seulement

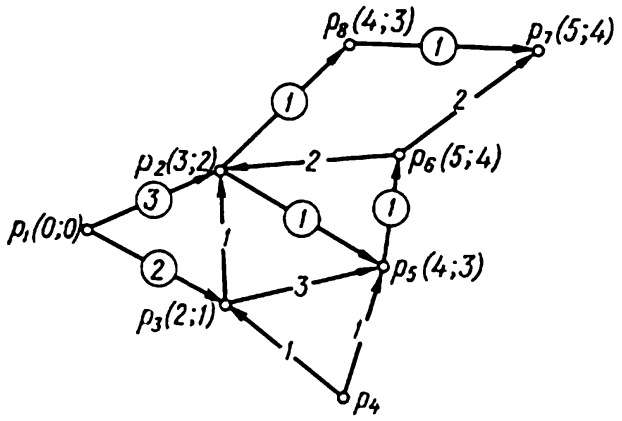

Fig. 1.8

ceux qui forment l'ensemble $P' \subset P$. La procédure n'est poursuivie alors que jusqu'au moment où tous les points de l'ensemble P' soient marqués ou bien jusqu'à ce que soit formé l'ensemble complet des points marqués.

2. L'algorithme est adapté à la construction d'un seul itinéraire optimal entre les points p_1 et p_s. Or, dans plusieurs cas, il est utile de mettre en évidence tous les itinéraires optimaux allant de p_1 à p_s. On y parvient à l'aide d'une petite correction de l'algorithme. Si le minimum de la somme $h_\alpha + c_{\alpha\mu}$ s'obtient sur plusieurs communications se terminant en un même point, il faut alors dégager toutes les communications. On peut montrer (cf. exercice 2) qu'un itinéraire optimal quelconque est constitué de communications dégagées. Par suite, la construction de tous les itinéraires optimaux entre p_1 et p_s se ramène à la composition de tous les itinéraires du sous-réseau de transport (P, K_1) qui relient p_1 et p_s, où K_1 est l'ensemble des communications dégagées.

2.3. Nous illustrerons l'algorithme décrit de résolution du problème de l'itinéraire le moins cher par un exemple de réseau de transport (fig. 1.8).

Les valeurs c_{ij} du coût de transport des communications sont indiquées sur le dessin ; de plus, les communications sont marquées

par des ronds où figurent les c_{ij} correspondants. Les chiffres entre parenthèses portés près des notations des points désignent la valeur h_λ du coût de transport de l'itinéraire optimal reliant le point considéré à p_1 (premier chiffre) et le numéro de l'étape à laquelle le point a été marqué (deuxième chiffre).

La figure 1.8 traduit ainsi toute la marche de la résolution du problème.

A la première étape on a marqué le point p_3; à la deuxième, p_2. A chacune des deux étapes suivantes on marquait deux points: à la troisième p_8 et p_5, à la quatrième p_6 et p_7. Voici, par exemple, la description de la troisième étape. L'ensemble des points marqués aux deux premières étapes comprend p_1, p_2 et p_3. Les communications à examiner sont k_{28}, k_{25}, k_{35}. Calculons les sommes qui leur correspondent:

$$h_2 + c_{28} = 4,$$
$$h_2 + c_{25} = 4,$$
$$h_3 + c_{35} = 5.$$

La valeur minimale de la somme est 4; nous marquons les points p_8 et p_5 en portant près d'eux les chiffres 4 et 3 (numéro de l'étape); les communications à dégager sont k_{28} et k_{25}.

L'ensemble complet des points marqués s'obtient en quatre étapes et se compose de tous les points du réseau sauf p_4. Par conséquent, tout point du réseau, à l'exception de p_4, peut être réuni à p_1 par un itinéraire orienté issu de p_1. Les itinéraires optimaux se construisent à l'aide des communications dégagées. Par exemple, l'itinéraire optimal associé au point p_6 passe par les points p_5 et p_2.

2.4. L'algorithme du problème de l'itinéraire le moins cher exposé dans le présent paragraphe a été proposé presque simultanément par Bellman [6] et Shimbell [83]. Fondé sur les idées de la programmation dynamique, il est conçu pour déterminer simultanément des itinéraires optimaux entre deux points quelconques d'un réseau de transport (P, K).

Ecrivons la fonction du coût de transport $c(k_{ij})$, $k_{ij} \in K$, sous la forme d'une matrice carrée C dont l'ordre est égal au nombre N des points du réseau. Rappelons que le nombre

$$c_{ij} = \begin{cases} c(k_{ij}) \text{ si } i \neq j \text{ et } k_{ij} \in K, \\ 0 \text{ si } i = j; \\ \infty \text{ si } i \neq j \text{ et } k_{ij} \notin K \end{cases}$$

est porté à l'intersection de la i-ème ligne et de la j-ième colonne de la matrice C.

Convenons de dire qu'un itinéraire composé de s communications a une longueur s. Appelons *s-optimal* un itinéraire orienté de p_i vers p_j de longueur inférieure ou égale à s, associé au coût minimal

par rapport à tous les autres itinéraires orientés de p_i vers p_j de longueur inférieure ou égale à s. Selon ces notations, les éléments de la matrice C sont les valeurs du coût de transport des itinéraires 1-optimaux.

Introduisons dans la discussion la matrice $C^{(s)} = \| c_{ij}^{(s)} \|_N$ du coût de transport des itinéraires s-optimaux. Comme d'habitude, la diagonale de $C^{(s)}$ est composée de zéros, tandis que les éléments infinis indiquent l'absence d'itinéraire de longueur inférieure ou égale à s reliant les points respectifs. D'après la notation, $C = C^{(1)}$. L'algorithme a à sa base la relation simple

$$c_{ij}^{(m+n)} = \min_{1 \leqslant \lambda \leqslant N} (c_{i\lambda}^{(m)} + c_{\lambda j}^{(n)}), \tag{2.4}$$

qui permet de calculer les éléments de la matrice $C^{(m+n)}$ d'après les éléments des matrices $C^{(m)}$ et $C^{(n)}$. Pour justifier l'égalité (2.4), constatons que l'itinéraire composé de l'itinéraire m-optimal de p_i à p_λ et de l'itinéraire n-optimal de p_λ à p_j a une longueur ne dépassant pas $m + n$. Par conséquent,

$$c_{ij}^{(m+n)} \leqslant \min_{1 \leqslant \lambda \leqslant N} (c_{i\lambda}^{(m)} + c_{\lambda j}^{(n)}). \tag{2.5}$$

Supposons que l'itinéraire $(m + n)$-optimal de p_i à p_j passe par le point p_α et qu'en outre la longueur de sa partie entre p_i et p_α ne dépasse pas m alors que la longueur de la partie entre p_α et p_j soit inférieure ou égale à n. Dans ce cas

$$c_{ij}^{(m+n)} \geqslant c_{i\alpha}^{(m)} + c_{\alpha j}^{(n)}. \tag{2.6}$$

En comparant (2.5) et (2.6) on obtient (2.4). Puisque la longueur de tout itinéraire d'un réseau de transport (P, K) ne dépasse pas $N - 1$, où N est le nombre de points de l'ensemble P, avec $s \geqslant N - 1$

$$C^{(s)} = C^*. \tag{2.7}$$

Par C^* on entend ici la matrice du coût de transport de différents itinéraires optimaux. L'égalité (2.7) peut avoir lieu également pour de plus petites valeurs du paramètre s.

L'algorithme de recherche des itinéraires optimaux pour tous les couples de points du réseau consiste à construire une suite de matrices $\{C^{(s_k)}\}_k$ dont le dernier élément est C^*. Ensuite, la matrice C^* permet de déterminer facilement chacun des itinéraires optimaux.

D'après la formule (2.4)

$$c_{ij}^{(2k)} = \min_{1 \leqslant \lambda \leqslant N} (c_{i\lambda}^{(k)} + c_{\lambda j}^{(k)}). \tag{2.8}$$

La relation (2.8) rend possible la composition successive des matrices

$$C^{(1)},\ C^{(2)},\ C^{(4)},\ C^{(8)},\ \ldots,\ C^{(2^k-1)},\ C^{(2^k)},\ \ldots \tag{2.9}$$

C'est pour tâcher d'obtenir la croissance la plus rapide possible du paramètre s que l'on fait appel précisément à une telle suite de matrices $C^{(s)}$.

Le calcul des éléments de la suite (2.9) se poursuit jusqu'à ce que l'on obtienne pour un certain d deux matricos voisines égales

$$C^{(2^d)} = C^{(2^{d+1})}. \tag{2.10}$$

Il est évident qu'avec $s = 2^d$ l'égalité (2.10) est équivalente à (2.7) (cf. exercice 5). Ainsi la matrice C^* s'obtient au bout d'un nombre d'étapes inférieur ou égal à $[\log_2 (N-1)] + 1$ *), chacune des étapes consistant à construire une matrice $C^{(2^k)}$ d'après la matrice connue $C^{(k)}$ selon la formule (2.8). Lorsque la matrice C^* est obtenue, on compose sans peine n'importe quel itinéraire optimal.

Soient p_i et p_j des points quelconques d'un réseau de transport.

Si $c^*_{ij} = \infty$, il n'existe aucun itinéraire commençant en p_i et se terminant en p_j.

Si par contre $c^*_{ij} < \infty$, de tels itinéraires existent, et pour construire le moins cher on procède de la façon suivante. Trouvons l'indice λ_1 à partir de la condition

$$c_{i\lambda_1} + c^*_{\lambda_1 j} = \min_{\lambda \in E_i} (c_{i\lambda} + c^*_{\lambda j}),$$

où E_i est l'ensemble des extrémités des communications $k_{i\lambda}$ à origine en p_i. Après p_i l'itinéraire optimal passe par p_{λ_1}. Les autres points de l'itinéraire à établir sont recherchés de façon analogue.

Si les points $p_{\lambda_1}, p_{\lambda_2}, \ldots, p_{\lambda_t}$ sont déjà trouvés, le point suivant $p_{\lambda_{t+1}}$ est donné par la condition

$$c_{\lambda_t \lambda_{t+1}} + c^*_{\lambda_{t+1} j} = \min_{\lambda \in E_{\lambda_t}} (c_{\lambda_t \lambda} + c^*_{\lambda j}). \tag{2.11}$$

La procédure se poursuit tant qu'avec un certain s le point $p_{\lambda_{s+1}}$ se trouve identique au point terminal p_j. L'itinéraire optimal cherché passe successivement par les points

$$p_i,\ p_{\lambda_1},\ p_{\lambda_2},\ \ldots,\ p_{\lambda_{s-1}},\ p_{\lambda_s},\ p_j.$$

Nous laissons au lecteur le soin de démontrer le bien-fondé du procédé exposé de construction des itinéraires optimaux d'après la matrice C^* (cf. exercice 6).

Il est possible d'obtenir à volonté tous les itinéraires optimaux allant de p_i à p_j. Dans ce but il faut pour tout t marquer non pas

*) $[X]$ est la partie entière du nombre X.

un seul, mais tous les indices λ_{t+1} qui vérifient la condition (2.11). Naturellement, cette opération doit être effectuée pour tous les λ_t déjà marqués.

Si $\lambda'_1, \lambda'_2, \ldots, \lambda'_s$ est une suite quelconque, composée d'indices marqués respectifs,

$$p_t,\ p'_{\lambda_1},\ \ldots,\ p'_{\lambda_{s-1}},\ p'_{\lambda_s},\ p_j$$

est un itinéraire optimal.

2.4. Voici un exemple numérique. Supposons qu'il faille déterminer tous les itinéraires optimaux du réseau de transport représenté figure 1.9.

Les segments qui correspondent aux communications ne portent pas de flèches. Cela signifie que les points rattachés par un segment sont reliés par deux communications opposées; le coût de transport de ces deux communications est le même. Composons la matrice du coût de transport des communications:

$$C = C^{(1)} = \begin{Vmatrix} 0 & 1 & \infty & 2 & 2 \\ 1 & 0 & 1 & \infty & 1 \\ \infty & 1 & 0 & 2 & 3 \\ 2 & \infty & 2 & 0 & 1 \\ 2 & 1 & 3 & 1 & 0 \end{Vmatrix}.$$

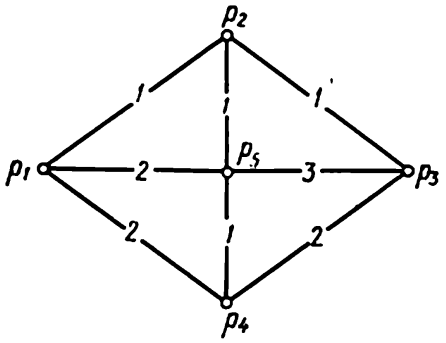

Fig. 1.9

La relation (2.8) avec $k = 1$ conduit à la matrice

$$C^{(2)} = \begin{Vmatrix} 0 & 1 & 2 & 2 & 2 \\ 1 & 0 & 1 & 2 & 1 \\ 2 & 1 & 0 & 2 & 2 \\ 2 & 2 & 2 & 0 & 1 \\ 2 & 1 & 2 & 1 & 0 \end{Vmatrix}.$$

Effectuons encore une étape pour calculer la matrice $C^{(4)}$. On vérifie aisément que $C^{(2)} = C^{(4)}$. Par conséquent, la matrice $C^{(2)}$ coïncide avec la matrice C^* du coût de transport des itinéraires optimaux.

La matrice $C^* = C^{(2)}$ permet de construire sans peine n'importe quel itinéraire optimal. Supposons, par exemple, qu'il faille trouver un itinéraire optimal entre p_1 et p_3.

$$\min\,(c_{12} + c^*_{23},\ c_{15} + c^*_{53},\ c_{14} + c^*_{43}) = \min\,(2, 4, 4) = 2.$$

p_2 fait donc partie de l'itinéraire cherché.

Ensuite,

$$\min\,(c_{23} + c^*_{33},\ c_{25} + c^*_{53}) = \min\,(1, 3) = 1,$$

ce qui signifie que le point p_2 de l'itinéraire optimal est suivi par le point p_3. L'itinéraire optimal de p_1 à p_3 passe donc par le point intermédiaire p_2.

Indiquons une modification de la méthode exposée, de réalisation très commode sur calculateur digital.

En calculant la matrice $C^{(2k)}$ il faut, d'après la relation (2.8), mémoriser la matrice $C^{(k)}$. C'est pourquoi l'ordinateur doit conserver en mémoire deux matrices à chaque étape.

La simple considération qui suit permet de réduire la charge de la mémoire tout en améliorant la convergence de la méthode.

Après le calcul de l'élément successif de la nouvelle matrice, il faut le substituer à l'élément correspondant de l'ancienne matrice. L'élément suivant de la nouvelle matrice se calcule d'après l'ancienne matrice ainsi modifiée, etc.

Il est clair que la procédure de résolution du problème porte maintenant sur une matrice révisée par éléments. Ayant renouvelé tous les éléments de la matrice, on revient au début pour reprendre le cycle de révision. La procédure se poursuit tant que les transformations ultérieures cessent de modifier les éléments de la matrice. La matrice ainsi composée est constituée des valeurs des coûts de transport des itinéraires optimaux; autrement dit, elle coïncide avec la matrice C^*. Appelons comme auparavant étape de l'algorithme l'ensemble des opérations nécessaires pour réviser complètement la matrice. Si la matrice $\widetilde{C}$ se forme à partir de $C^{(k)}$ en une seule étape, il est évident que les éléments $\widetilde{c}_{ij}$ sont les valeurs du coût de transport des itinéraires $t\ (i,\ j)$-optimaux avec $t\ (i,\ j) \geqslant 2k$. De plus, plus la révision de l'élément $c_{ij}^{(k)}$ vient tard et plus il est probable que $t\ (i,\ j)$ soit supérieur à $2k$. Il en résulte que la modification considérée mène à la solution au moins aussi vite que la méthode principale. Dans la majorité des cas il se trouve que la convergence est plus rapide. Dans la mesure où les éléments de la matrice examinés en fin d'étape convergent généralement plus vite que les premiers éléments, il est sensé de changer après chaque étape l'ordre de révision des éléments. Ceci mène à une convergence plus uniforme du processus, ce qui à son tour permet d'obtenir plus vite la matrice cherchée C^*.

2.5. Dans la description des méthodes de construction des itinéraires optimaux, on supposait que les valeurs de la fonction C du coût de transport étaient non négatives. C'est une contrainte importante. On l'a appliquée pour déduire l'inégalité (2.2) qui est à la base de la méthode de Minty, ainsi que pour démontrer la formule (2.4), qui régit la méthode de Bellman-Shimbell. Comme le montrent les exemples, si certains c_{ij} sont négatifs la méthode de Minty de même que celle de Bellman-Shimbell ne mènent pas en général au but recherché. Plus même, il n'existe actuellement aucun algorithme valable pour construire un itinéraire optimal susceptible de conduire au but cherché avec $c_{ij} < 0$, bien que de nombreux problèmes importants de grand intérêt se ramènent justement à ce cas. Néanmoins dans certaines situations la méthode de Bellman-Shimbell peut être utile même pour des c_{ij} négatifs.

Nous avons déjà dit que l'image géométrique d'un itinéraire orienté est une ligne brisée qui ne se coupe pas elle-même. Etendons la notion d'itinéraire orienté en levant l'hypothèse sur l'absence d'intersections de la ligne brisée correspondante avec

elle-même. Notamment, appelons la suite des communications

$$k_{ii_1},\ k_{i_1i_2},\ \ldots,\ k_{i_{s-1}j}$$

itinéraire généralisé entre p_i et p_j.

La définition d'un itinéraire s-optimal (§ 2.3) se transfère naturellement à la classe plus large des itinéraires généralisés. Par $C^{(s)} = \| c_{ij}^{(s)} \|_N$ nous allons entendre la matrice du coût de transport des itinéraires généralisés s-optimaux. Des raisonnements parfaitement analogues à ceux du § 2.4 permettent d'établir que la formule (2.4) est vraie quels que soient les signes des éléments de la matrice $C = C^{(1)}$. Par conséquent, la méthode de Bellman-Shimbell est applicable quel que soit le cas de la recherche des itinéraires généralisés s-optimaux. Si le réseau (P, K) et la fonction $c(k_{ij})$ définie sur l'ensemble des communications K sont tels qu'un itinéraire généralisé s-optimal est un itinéraire ou bien peut être facilement transformé en un itinéraire s-optimal aux mêmes points extrémaux, cette méthode conduit évidemment au choix des itinéraires optimaux. Il en est ainsi, par exemple, lorsque $c(k_{ij}) \geqslant 0$. C'est précisément ce qui explique la simplicité relative du problème de l'itinéraire le moins cher dans le cas des fonctions C non négatives. Soit maintenant $c(k_{ij})$ une fonction quelconque et (P, K) le réseau de transport sans itinéraires fermés orientés. (Un exemple peut en être donné par des réseaux de transport construits en employant la méthode PERT). Sous cette hypothèse, tout itinéraire généralisé du réseau est également son itinéraire et par suite, la méthode de Bellman-Shimbell donne la solution du problème de l'itinéraire optimal. Les hypothèses exposées ci-dessus peuvent être remplacées par une contrainte plus faible.

Pour rendre possible l'application de la méthode de Bellman-Shimbell, il suffit d'imposer d'être non négatives aux valeurs du coût de transport de chacun des itinéraires fermés orientés du réseau.

§ 3. Problème du flot maximal

3.1. Soit un réseau de transport quelconque (P, K) et une fonction des capacités $d(k_{ij}) \geqslant 0$ définie sur l'ensemble K.

Fixons un certain couple de points du réseau (P, K). Numérotons les points de l'ensemble P de façon que le premier point du couple fixé (entrée) soit désigné p_1 et le deuxième (sortie) p_N, où N est le nombre total de points du réseau.

Considérons tous les flots possibles de p_1 à p_N compatibles avec la fonction des capacités donnée. Le problème consiste à rechercher parmi eux un flot $X^* = x^*(k_{ij})$, $k_{ij} \in K$, à valeur maximale x_1^* qui s'appelle flot *maximal* entre p_1 et p_N.

Nous allons voir par la suite que l'intérêt du problème du flot maximal ne réside pas seulement en lui-même. Dans certains cas

l'analyse de problèmes de réseaux plus complexes impose elle aussi la construction de flots maximaux sur les réseaux, composés au cours de la discussion du problème correspondant. Ce paragraphe donne l'exposé d'un algorithme qui permet de construire des flots maximaux pour des réseaux de transport quelconques. Cet algorithme a été établi par les mathématiciens américains L. R. Ford et D. R. Fulkerson [74], [75].

3.2. Avant de passer à la description de l'algorithme, signalons quelques propriétés simples mais utiles des flots dans les réseaux.

Il est plus commode d'exposer les raisonnements ci-dessous en termes des flots algébriques. C'est pourquoi dans ce qui suit nous comprendrons par flot une fonction $x(k_{ij})$, $k_{ij} \in K'$, vérifiant les conditions

$$x(k_{ij}) = -x(k_{ji}), \tag{3.1}$$

$$\sum_{k_{ij} \in E_i} x(k_{ij}) = 0 \text{ pour } i \neq 1, N, \tag{3.2}$$

$$x(k_{ij}) \leqslant d(k_{ij}), \quad k_{ij} \in K'. \tag{3.3}$$

Le sens des symboles E_i et K' a été expliqué au § 1.4. Il est commode d'étendre les fonctions $x(k_{ij})$, $k_{ij} \in K'$, et $d(k_{ij})$, $k_{ij} \in K'$, à l'ensemble de tous les couples ordonnés des points du réseau, en posant selon l'usage

$$d(k_{ij}) = x(k_{ij}) = 0,$$

si p_i et p_j ne sont reliés par aucune communication de l'ensemble K. Cette circonstance, ainsi que les notations introduites au début du § 1.4, permettent d'écrire l'identité

$$\sum_{k_{ij} \in E_i} x(k_{ij}) = x(p_i, P). \tag{3.4}$$

La valeur x_1 du flot X est donc déterminée par la relation

$$x_1 = x(p_1, P) = -x(p_N, P), \tag{3.5}$$

et les conditions (3.2) peuvent être récrites sous la forme

$$x(p_i, P) = 0 \text{ pour } i \neq 1, N. \tag{3.6}$$

Si P' est une partie quelconque de l'ensemble P, il vient

$$x(P', P) = 0. \tag{3.7}$$

L'égalité (3.7) est une conséquence de la condition (3.1) et de la présence dans le premier membre de (3.7) des termes $x(k_{ij})$ et $x(k_{ji})$ pourvu que $p_i, p_j \in P'$.

Supposons que l'ensemble des points P soit découpé en deux parties disjointes P' et P'', telles que $p_1 \in P'$ et $p_N \in P''$.

On a la chaîne d'égalités suivante:

$$x_1 = x(p_1, P) = \sum_{p_i \in P'} x(p_i, P) = x(P', P) + x(P', P'') = x(P', P'').$$

La première égalité de la chaîne est déjà établie. La deuxième se déduit de (3.6) et de l'hypothèse suivant laquelle $p_N \notin P'$. Les troisième et quatrième sont des conséquences de la propriété (1.3), et la cinquième est déterminée par la relation (3.7). Ainsi quels que soient le découpage de l'ensemble P en parties disjointes P' et P'' ($p_1 \in P'$, $p_N \in P''$) et le flot arbitraire $X = x(k_{ij})$, l'égalité

$$x_1 = x(P', P'') \tag{3.8}$$

est vérifiée.

Soient P' et P'' deux parties disjointes quelconques de l'ensemble P dont la somme est P et dont la première contient l'entrée p_1 et la deuxième la sortie p_N. Appelons l'ensemble $\Gamma \subset K'$ des communications issues des points de P' et finissant aux points de P'' *coupe* du réseau de transport (P, K') avec l'entrée p_1 et la sortie p_N. La définition univoque d'une coupe Γ est donnée par l'un des ensembles du couple P', P''.

Par *capacité* $d(\Gamma)$ de la coupe Γ on entend la capacité totale de toutes les communications constituant Γ

$$d(\Gamma) = \sum_{k_{ij} \in \Gamma} d(k_{ij}). \tag{3.9}$$

Si Γ est une coupe, tout itinéraire orienté de p_1 à p_N comprend au moins une communication de l'ensemble Γ. Considérons, en effet, un itinéraire arbitraire de p_1 à p_N qui passe par les points $p_1 = p_{i_1}, p_{i_2}, \ldots, p_{i_s} = p_N$. Soit λ le numéro minimal tel que $p_{i_\lambda} \in P''$. Alors, la communication $k_{i_{\lambda-1} i_\lambda} \in \Gamma$, car $p_{i_{\lambda-1}} \in P'$.

Soit $x(k_{ij})$, $k_{ij} \in K'$, un flot quelconque du réseau (P, K') compatible avec la fonction $d(k_{ij})$, $k_{ij} \in K'$, et Γ une coupe arbitraire de ce réseau.

On a l'inégalité

$$x_1 \leqslant d(\Gamma). \tag{3.10}$$

En effet, si P' et P'' sont les parties de l'ensemble P qui ont engendré la coupe Γ, d'après (3.8), (3.3) et (3.9)

$$x_1 = x(P', P'') = \sum_{k_{ij} \in \Gamma} x(k_{ij}) \leqslant \sum_{k_{ij} \in \Gamma} d(k_{ij}) = d(\Gamma).$$

L'inégalité (3.10) définit le critère suivant d'optimalité d'un flot: *si la valeur x_1^* du flot X_1^* est égale à la capacité d'une certaine coupe du réseau, X^* est alors un flot maximal.*

3.3. Nous sommes maintenant à même d'exposer l'algorithme de construction d'un flot optimal dans un réseau quelconque (P, K)

à entrée p_1 et à sortie p_N, compatible avec une certaine fonction des capacités $d(k_{ij})$, $k_{ij} \in K$.

Cet algorithme se compose d'itérations distinctes dont chacune établit si le flot disponible est maximal, ou bien s'il faut construire un nouveau flot de valeur supérieure à celle de l'ancien flot.

Donnons la description d'une itération de l'algorithme, en supposant connu un certain flot $X = x(k_{ij})$, $k_{ij} \in K'$. Commençons par les définitions. Soit $X = x(k_{ij})$, $k_{ij} \in K'$, un flot quelconque compatible avec la fonction $d(k_{ij})$, $k_{ij} \in K'$. La communication $k_{ij} \in K'$ est dite *X-saturée* si $x_{ij} = x(k_{ij}) = d_{ij}$; la communication k_{ij} est dite *X-insaturée* si $x_{ij} < d_{ij}$. Ces définitions supposent que X est un flot algébrique. Par conséquent, l'ensemble des communications X-insaturées est composé des communications $k_{ij} \in K$ telles que $x_{ij} < d_{ij}$, et des communications $k_{ij} \notin K$ telles que $x_{ji} > 0$. Les autres communications constituent l'ensemble des communications X-saturées.

Une itération consiste à construire de proche en proche un ensemble de points marqués du réseau. On commence par marquer l'entrée p_1. Pendant l'exécution de l'itération des éléments toujours nouveaux viennent s'inclure dans l'ensemble des points marqués.

Appelons *étape de l'itération* l'ensemble des opérations qui augmentent de un le nombre de points marqués ou bien qui établissent que l'ensemble de ces points est complètement défini. Supposons que nous ayons déjà effectué t étapes et obtenu l'ensemble P_t des points marqués du réseau. La $(t+1)$-ième étape consiste à réviser d'une façon régulière les points de l'ensemble P_t dans le but de révéler le point marqué formant l'origine d'une communication X-insaturée qui se termine en un point non marqué. Lorsqu'un tel point p_λ est trouvé, on attribue au point non marqué p_μ, extrémité de la communication X-insaturée $k_{\lambda\mu}$, le nombre λ, ce qui met fin à la $(t+1)$-ième étape.

Deux cas peuvent alors se présenter:

a) le point p_μ n'est pas la sortie p_N;

b) le point p_μ est la sortie p_N.

Dans le cas a), nous passons à l'étape suivante de l'itération en posant $P_{t+1} = \{P_t, p_\mu\}$.

Le cas b) indique que P_{t+1} est l'ensemble complet des points marqués.

Il se peut qu'après avoir revu tous les points de l'ensemble P_t, nous constations qu'aucun point p_λ ne soit relié au point non marqué p_μ par une communication X-insaturée $k_{\lambda\mu}$. Appelons cette éventualité cas c). Le cas c) signifie que P_t est l'ensemble complet des points marqués. Ainsi, l'étape d'une itération peut avoir une des trois issues: a) impose l'exécution de l'étape suivante, b) et c) fixent la construction de l'ensemble complet des points marqués que nous désignerons par P'. Chaque étape de l'itération (sauf la dernière

peut-être) accroît le nombre de points marqués. C'est pourquoi, après un nombre fini d'étapes nous tombons nécessairement sur le cas b) ou c).

3.4. Examinons séparément chacune de ces trois éventualités. Le cas b) consiste à construire l'ensemble complet des points marqués P', qui comprend la sortie p_N.

Montrons qu'il en résulte la possibilité de construire un flot $X' = x'(k_{ij})$, $k_{ij} \in K'$, dont la valeur x_1' est plus grande que la valeur x_1 du flot X.

Formons la suite $\{p_{i_\alpha}\}$ des points de l'ensemble P' en se guidant sur les règles qui suivent.

Posons $p_{i_1} = p_N$. Si les points $p_{i_1}, p_{i_2}, \ldots, p_{i_{\alpha-1}}$ de la suite sont déjà obtenus, l'indice i_α qui définit le α-ième point de la suite est considéré égal au nombre attribué au point $p_{i_{\alpha-1}}$. La composition de la suite se poursuit tant que cela est possible.

Chaque point de l'ensemble P', à l'exception de l'entrée p_1, est affecté d'un certain nombre. Tous les éléments de la suite sont différents, car l'hypothèse contraire entraîne l'existence d'un point marqué deux fois.

La suite $\{p_{i_\alpha}\}$ contient donc un nombre fini de points et se termine par l'entrée p_1, c'est-à-dire prend la forme

$$p_N = p_{i_1},\ p_{i_2},\ p_{i_3},\ \ldots,\ p_{i_{s-1}},\ p_{i_s} = p_1. \tag{3.11}$$

D'après la règle d'extension des ensembles de points marqués, les communications

$$k_{i_{\lambda+1},\, i_\lambda} \text{ avec } 1 \leqslant \lambda \leqslant s-1$$

sont X-insaturées, c'est-à-dire

$$x_{i_{\lambda+1},\, i_\lambda} < d_{i_{\lambda+1},\, i_\lambda} \text{ pour } \lambda = 1,\ 2,\ \ldots,\ s-1. \tag{3.12}$$

Posons

$$\theta = \min_{1 \leqslant \lambda \leqslant s-1} (d_{i_{\lambda+1},\, i_\lambda} - x_{i_{\lambda+1},\, i_\lambda}), \tag{3.13}$$

$$x'(k_{ij}) = -x'(k_{ji}) =$$

$$= \begin{cases} x(k_{ij}) + \theta \text{ si } k_{ij} = k_{i_{\lambda+1},\, i_\lambda} \text{ pour } \lambda = 1,\ 2,\ \ldots,\ s-1, \\ x(k_{ij}) \text{ dans le cas contraire.} \end{cases} \tag{3.14}$$

D'après les relations (3.12) et (3.13)

$$\theta > 0, \tag{3.15}$$

on suppose en outre que $\theta < \infty$.

La fonction $x'(k_{ij})$, $k_{ij} \in K'$, est un flot algébrique compatible avec la fonction des capacités $d(k_{ij})$, $k_{ij} \in K'$.

En effet, les conditions (3.2) ne peuvent être compromises qu'avec $i = i_\lambda$ pour $\lambda = 2, 3, \ldots, s-1$. Mais selon (3.14), pour une telle valeur de λ (quelle que soit cette valeur) la composante $x_{i_\lambda, i_{\lambda-1}}$ du flot $x(k_{ij})$ augmente de θ, la composante $x_{i_\lambda, i_{\lambda+1}} = -x_{i_{\lambda+1}, i_\lambda}$ diminue de la même valeur θ, alors que les autres composantes du flot associées aux communications issues de p_λ gardent leurs valeurs précédentes. Par conséquent, les conditions (3.2) sont également vraies pour $i = i_\lambda$.

Les conditions (3.3) ne sont pas compromises non plus, puisque d'après (3.14) et (3.15)

$$x'(k_{i_{\lambda+1}, i_\lambda}) = x(k_{i_{\lambda+1}, i_\lambda}) + \theta \leqslant x(k_{i_{\lambda+1}, i_\lambda}) + d_{i_{\lambda+1}, i_\lambda} - \\ - x(k_{i_{\lambda+1}, i_\lambda}) = d_{i_{\lambda+1}, i_\lambda};$$

$$x'(k_{i_\lambda, i_{\lambda+1}}) = -x'(k_{i_{\lambda+1}, i_\lambda}) = -x(k_{i_{\lambda+1}, i_\lambda}) - \\ - \theta < x(k_{i_\lambda, i_{\lambda+1}}) \leqslant d_{i_\lambda, i_{\lambda+1}}.$$

Estimons la valeur x_1' du flot X' nouvellement construit. Parmi les composantes $x(k_{1j})$ du flot X, seule $x(k_{1i_{s-1}})$ a été modifiée en augmentant sa valeur de $\theta > 0$. Donc

$$x_1^{(1)} = x_1 + \theta > x_1. \tag{3.16}$$

Ainsi, si le résultat de l'étape successive est b), on construit un nouveau flot dont la valeur est supérieure à celle de l'ancien.

Considérons maintenant le cas c). Ce cas consiste dans le fait que l'ensemble construit P' des points marqués, en premier lieu, ne contient pas de sortie p_N, et puis ne peut pas subir d'extension. Montrons que l'existence du cas c) indique que le flot $x(k_{ij})$ est maximal.

Soit P'' l'ensemble des points non marqués du réseau. Il est clair que P' et P'' n'ont pas de points communs et que leur union forme l'ensemble P. De plus,

$$p_1 \in P', \quad p_N \in P''.$$

Par suite, l'ensemble Γ des communications allant des points P' aux points P'' est une coupe du réseau (P, K').

Si k_{ij} est une communication quelconque de la coupe Γ, on a

$$x(k_{ij}) = d_{ij}. \tag{3.17}$$

En effet, le raisonnement par l'absurde rendrait possible l'extension de l'ensemble P' auquel on aurait adjoint un point p_j non encore marqué.

Les égalités (3.17) et (3.8) entraînent

$$d(\Gamma) = \sum_{k_{ij} \in \Gamma} d(k_{ij}) = \sum_{k_{ij} \in \Gamma} x(k_{ij}) = x(P', P'') = x_1. \tag{3.18}$$

Les relations (3.10) et (3.18) conduisent à l'inégalité $x_1 \leqslant \bar{x}_1$, où $\bar{x}_1$ est la valeur d'un flot arbitraire $\bar{X}$ compatible avec la fonction D.

Par conséquent $x(k_{ij})$ est le flot maximal recherché.

Ainsi, lorsqu'une étape de l'itération se termine par le résultat c), le flot $x(k_{ij})$ est maximal.

3.5. Résumons le contenu des deux paragraphes précédents. L'itération d'un algorithme consiste en un nombre fini d'étapes qui conduisent à la construction de l'ensemble complet des points marqués P'. Le résultat de la dernière étape de l'itération soit impose une amélioration du flot disponible [cas b)] soit montre qu'il est maximal [cas c)]. Dans le premier cas nous dirons résultat 1 de l'itération considérée si $\theta < \infty$, et dans le deuxième, résultat 2. Le résultat 3 de l'itération est enregistré lorsque $\theta = \infty$, c'est-à-dire lorsqu'on construit entre p_1 et p_N un itinéraire dont chaque communication a une capacité illimitée. Le résultat 3 témoigne de l'existence des flots compatibles avec la fonction D et dont les valeurs sont aussi grandes que l'on veut.

L'algorithme se compose de plusieurs itérations successives et comporte une partie préalable et une partie principale.

La première itération s'effectue en partant du flot nul, dont toutes les composantes sont des zéros. Chaque itération successive est réalisée en partant du flot construit lors de l'itération précédente. On ne s'adresse à l'itération suivante que lorsque l'itération précédente aboutit au résultat 1.

Au début on procède aux itérations de la partie préalable de l'algorithme. Avant de s'attaquer à chacune de ces itérations on élimine *) du réseau (P, K') les communications déjà saturées à l'itération précédente. Les itérations de la partie auxiliaire de l'algorithme se poursuivent tant que l'une d'elles amène le résultat 2 ou 3. Le nombre fini d'itérations nécessaire à cet effet ne dépasse pas évidemment le nombre total des communications du réseau.

Si l'on obtient le résultat 3, le problème du flot maximal n'a pas de solution et il faut mettre fin à la procédure. Le résultat 2 indique qu'il est nécessaire de s'adresser à la partie principale de l'algorithme, mais en restituant, avant de la commencer, toutes les communications éliminées lors des itérations précédentes. Durant toutes les itérations de la partie principale de l'algorithme, l'ensemble des communications reste invariable. Supposons que les capacités d_{ij} de toutes les communications soient des entiers. Nous allons montrer que le flot maximal cherché s'obtient en un nombre fini d'itérations.

Si la partie préalable de l'algorithme s'achève par le résultat 2, les valeurs des flots compatibles avec la fonction D sont bornées

*) Les communications éliminées ne participent pas aux itérations ultérieures de la partie auxiliaire de l'algorithme: les composantes du flot sur ces communications gardent des valeurs invariables.

supérieurement. En effet, soit l un itinéraire quelconque composé de communications du réseau et allant de p_1 à p_N. On remarque sans peine que l se compose nécessairement de communications à capacité bornée. Le seul cas à vérifier est celui où l ne contient aucune des communications éliminées par la partie préalable de l'algorithme. Mais dans ce cas l est l'itinéraire du réseau exploré par la dernière itération de la partie préalable de l'algorithme; notre conclusion se déduit donc de l'existence d'un flot maximal de ce réseau, ce qui est conditionné à son tour par le résultat 2. Réunissons en un ensemble S' tous les points du réseau (P, K') qu'on peut atteindre depuis p_1 en suivant des itinéraires orientés composés de communications à capacité illimitée; les autres points du réseau seront réunis en un ensemble S''. Ce qui a été démontré amène $p_N \in S''$, et par suite, d'après (3.8), la valeur x_1 de n'importe quel flot X vérifie la relation

$$x_1 = x(S', S'') = \sum_{k_{ij} \in R} x(k_{ij}), \tag{3.19}$$

où R est l'ensemble des communications du réseau allant des points S' aux points S''.

Toutes les communications de l'ensemble R ont des capacités bornées, car s'il n'en est pas ainsi on pourrait adjoindre l'un des points de l'ensemble S'' à l'ensemble S'. On en tire que pour tout flot X compatible avec la fonction D, l'égalité (3.19) entraîne l'inégalité

$$x_1 = \sum_{k_{ij} \in R} x(k_{ij}) \leqslant \sum_{k_{ij} \in R} d(k_{ij}) = d < \infty, \tag{3.20}$$

qui témoigne de la valeur bornée de tous les flots de ce type. Notons ensuite que sous l'hypothèse des d_{ij} entiers, les composantes de tous les flots formés aux différentes itérations de l'algorithme sont des entiers. C'est pourquoi toute itération qui aboutit au résultat 1 accroît la valeur du flot d'un entier supérieur à zéro, c'est-à-dire d'une unité au moins. Ce fait conduit avec la relation (3.20) à la conclusion que le nombre total d'itérations de l'algorithme ne peut pas dépasser le nombre d, la dernière itération devant nécessairement s'achever par le résultat 2.

Ainsi, ou bien la partie préalable de l'algorithme amène le résultat 3, ou bien sa partie principale s'achève par le résultat 2; de plus le nombre total d'itérations de l'algorithme est fini. Cette proposition peut être aisément étendue au cas où toutes les grandeurs d_{ij} sont des nombres rationnels (cf. exercice 7).

Si un flot maximal existe, l'algorithme considéré permet de l'obtenir en un nombre fini d'itérations. Sinon on constate également en un nombre fini d'itérations que le problème n'a pas de solution. Notons que lorsque l'existence d'un flot maximal est assurée, on peut ne pas diviser l'algorithme en parties préalable et principale.

La procédure de construction du flot maximal consiste alors en un nombre fini d'itérations de la partie principale de l'algorithme.

L'algorithme du flot maximal peut être utilisé non seulement pour des calculs numériques, mais encore comme un outil de recherche théorique.

Voici une proposition importante qui se déduit sans peine de l'algorithme exposé.

Disons qu'une coupe Γ^* du réseau (P, K') est *minimale* si

$$d(\Gamma^*) \leqslant d(\Gamma)$$

pour toutes les coupes Γ du réseau.

Théorème 3.1. *Si X^* est un flot maximal du réseau (P, K') de valeur x_1^*, il existe également une coupe minimale Γ^* de ce réseau de capacité $d(\Gamma^*)$, et de plus*

$$x_1^* = d(\Gamma^*). \tag{3.21}$$

Nous donnerons la démonstration sous l'hypothèse que tout d_{ij} est un entier. Appliquons au réseau (P, K') l'algorithme exposé plus haut et qui conduit après un nombre fini d'itérations au résultat 2 (le flot maximal existe).

Soient Γ^* la coupe définie par les ensembles P' et P'' construits à la dernière itération et X^* le flot maximal obtenu. D'après l'égalité (3.18),

$$x_1^* = d(\Gamma^*).$$

En utilisant ensuite l'inégalité (3.10) avec $x_1 = x_1^*$, on amène la relation

$$d(\Gamma^*) \leqslant d(\Gamma),$$

vraie pour n'importe quelle coupe Γ du réseau.

Donc, Γ^* est une coupe minimale. Ainsi la coupe minimale du réseau (P, K') existe et sa capacité est égale à la valeur du flot maximal. Le théorème est démontré.

La recherche de la coupe minimale est un problème dual par rapport à celui du flot maximal. On pourrait s'en convaincre en considérant le problème du flot maximal comme un problème de programmation linéaire qu'il constitue à l'évidence.

Le théorème 3.1 est le premier théorème de dualité du problème du flot maximal.

Constatons que l'algorithme du flot maximal donne non seulement la solution du problème direct, mais détermine aussi la coupe minimale du réseau.

3.6. Voici quelques remarques qui peuvent être utiles pour construire le schéma numérique de la résolution du problème du flot maximal. Nous avons vu qu'une itération de cette méthode consiste en principe à construire l'ensemble P' des points marqués. A cet effet il faut réviser à chaque étape les points marqués. L'ordre de

cette révision peut être arbitraire. Il faut seulement que tous les points marqués soient passés en revue. En se donnant tel ou tel ordre on aboutit aux réalisations numériques différentes de l'algorithme. On peut considérer, par exemple, que chaque point nouvellement marqué est le dernier dans l'ensemble des points marqués disponible. Une autre possibilité consiste à affecter le point nouvellement marqué d'un numéro supérieur d'une unité au numéro du dernier point révisé; les numéros de tous les points ultérieurs sont déplacés de l'unité. On peut proposer toute une série d'autres règles pour définir l'ordre de révision des points marqués.

La révision successive des points marqués est loin d'être le seul moyen possible de formation de l'ensemble P'. Il est clair qu'on peut aboutir au même résultat en examinant les points non marqués. Dès qu'on découvre un point non marqué relié par une communication X-insaturée à l'un des points marqués, on l'inclut dans l'ensemble des points marqués. L'ordre de révision des points non marqués peut être aussi des plus variés.

Il est parfois plus avantageux de construire un ensemble des points marqués (l'ensemble $\widetilde{P}'$) en partant non de l'entrée p_1 mais de la sortie p_N. On inclut successivement dans l'ensemble $\widetilde{P}'$ les points d'issue au moins d'une communication X-insaturée qui se termine en l'un des points déjà marqués. Le processus de construction de $\widetilde{P}'$ est considéré comme achevé si l'on marque à la dernière étape l'entrée p_1 ou si l'extension ultérieure de l'ensemble des points marqués s'avère impossible. Le premier cas permet de poursuivre l'amélioration du flot, le second traduit l'obtention du flot maximal.

L'itinéraire de p_1 à p_N composé de communications X-insaturées peut être recherché en partant simultanément de l'entrée et de la sortie. En d'autres termes, on peut construire simultanément deux ensembles de points marqués, celui des p_1 et celui des p_N. Dès que l'on obtient le point à inclure dans les deux ensembles, l'itinéraire recherché est trouvé et on peut améliorer le flot. La fin de la procédure de formation des ensembles des points marqués est fixée encore par l'impossibilité de poursuivre leur extension, ce qui indique que le flot disponible est maximal. Chacune des méthodes indiquées de construction d'un itinéraire non saturé de p_1 à p_N peut certes se calculer d'une façon différente en fonction de l'ordre choisi de révision des points marqués ou non du réseau.

D'après la formule (3.16), la valeur du flot construit au cours de l'itération donnée est supérieure à celle de l'ancien flot d'un nombre $\theta > 0$, et de plus, selon (3.13), le nombre θ est le minimum des écarts entre les composantes de l'ancien flot et les valeurs correspondantes de d_{ij} le long de l'itinéraire insaturé obtenu. Appelons le nombre θ *degré d'insaturation* de l'itinéraire. Dans le cas général, l'entrée et la sortie peuvent être reliées par un grand nombre d'itinéraires insaturés. Il est logique de rechercher une procédure de formation

d'un itinéraire insaturé telle qu'elle conduise à un itinéraire entre p_1 et p_N au degré d'insaturation le plus grand possible. La croissance de la valeur du flot à chaque itération sera alors maximale.

La mise en œuvre de cette remarque peut être réalisée à l'aide d'une modification de l'un des algorithmes de l'itinéraire le moins cher.

Choisissons à cette fin la méthode de Minty (cf. § 2.2). Chaque point marqué p_i (sauf l'entrée p_1) est affecté d'un indice supplémentaire θ_i égal au plus grand des degrés d'insaturation des itinéraires allant de p_1 à p_i. L'indice θ_i se calcule au moment de l'inclusion du point p_i dans l'ensemble des points marqués. Ceci devient possible grâce à la légère modification suivante de la règle du choix du point à inclure au nombre des points marqués. A chaque étape de l'itération la révision porte sur *tous* les points non marqués. On choisit parmi eux ceux des points où finit l'une des communications insaturées qui partent d'un point marqué.

Pour chaque point p_i de ce type on calcule le nombre

$$\delta_i = \max \{\min (\theta_\lambda, d_{\lambda i} - x_{\lambda i})\},$$

où le maximum est pris pour tous ceux des numéros λ des points marqués p_λ d'où sortent des communications insaturées se terminant en p_i.

On dégage ensuite parmi les points sélectionnés p_i le point $p_{i'}$ ayant la plus grande valeur du nombre $\delta_{i'}$. Ce point est inclus dans le nombre des points marqués, et l'on pose que $\theta_{i'}$ est égal à $\delta_{i'}$. Si max δ_i s'obtient pour plusieurs indices i, on marque alors tous les points correspondants en attribuant à chacun d'eux un nombre égal à max δ_i.

Comme auparavant, la procédure de formation de l'ensemble P' des points marqués s'achève soit par l'inclusion dans P' de la sortie p_N, soit par la constatation du fait que l'extension ultérieure de cet ensemble est impossible.

Dans le premier cas, la règle exposée dans la description d'une itération isolée permet de construire un itinéraire de p_1 à p_N constitué de communications insaturées. On voit sans peine que le degré d'insaturation de l'itinéraire ainsi construit est le plus grand possible (cf. exercice 9). Le deuxième cas indique que le flot disponible est optimal.

La modification exposée de l'algorithme du flot maximal augmente quelque peu le volume de calcul nécessaire pour l'exécution d'une itération, mais dans la majorité des cas elle diminue nettement le nombre total d'itérations nécessaires pour résoudre le problème.

Voici encore une remarque simple dont l'observation permet de réduire le volume de calcul à chaque itération. Lorsqu'on passe à l'itération suivante, le système des communications insaturées change. Il faut donc reprendre à chaque itération le processus de

formation de l'ensemble P'. Il est cependant facile d'indiquer un certain sous-ensemble de points marqués de l'itération précédente qui entre en entier dans l'ensemble P' pour l'itération successive. Dans le processus de formation de P' il est logique de prendre ce sous-ensemble pour ensemble initial.

Supposons que l'itération précédente se soit achevée par la construction d'un nouveau flot par modification de l'ancien flot le long de l'itinéraire qui passe par les points (3.11). Soit μ la valeur maximale des indices λ tels que

$$\theta = d_{i_{\lambda+1}, i_\lambda} - x_{i_{\lambda+1}, i_\lambda}.$$

Il est évident que le passage des communications insaturées aux communications saturées et inversement par suite de la modification du flot n'est possible que pour les communications reliant les points p_{i_λ} et $p_{i_{\lambda+1}}$ avec $\lambda \leqslant \mu$.

Par conséquent, tous les points qui furent marqués à l'itération précédente avant le point $p_{i_{\mu+1}}$ le seront encore dans le même ordre à l'itération suivante. Ces points restent dans le nouvel ensemble P' et, de plus, les indices qui leur sont affectés gardent leur valeur précédente.

3.7. Donnons un exemple d'utilisation de l'algorithme du flot optimal. Considérons le réseau de la figure 1.10, avec l'entrée p_1 et la sortie p_8. Les valeurs des capacités des communications isolées sont portées sur la même figure.

Nous diviserons la procédure de résolution en parties préalable et principale (on pourrait ne pas le faire, puisque le flot maximal existe d'avance et sa valeur ne dépasse pas $d_{12} + d_{13} + d_{14} = 12$).

Itération 1. Le flot initial est un flot nul. Les points

$$p_1,\ p_2\,(1),\ p_5\,(2),\ p_8\,(5)$$

sont inclus successivement dans l'ensemble des points marqués.

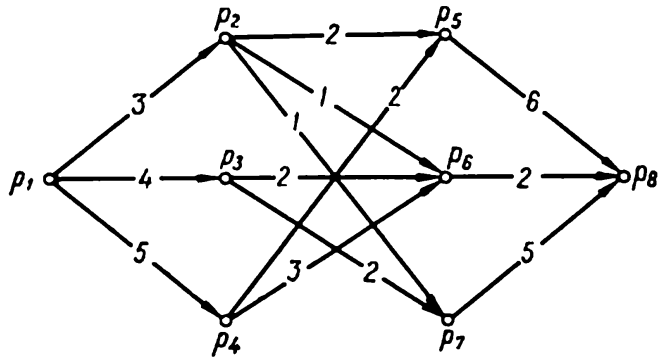

Fig. 1.10

Ici et dans ce qui suit on utilise un ordre de révision tel que le point nouvellement marqué reçoit le numéro d'ordre minimal disponible, c'est-à-dire qu'il est révisé en premier lieu.

L'itération compte trois étapes, la dernière s'achevant par le résultat b) (le point p_8 (la sortie) se range parmi les points marqués).

On passe au nouveau flot $X^{(1)}$ en posant

$$x_{12}^{(1)} = x_{25}^{(1)} = x_{58}^{(1)} = \theta = \min(3,\ 2,\ 6) = 2.$$

Les autres composantes du flot $X^{(1)}$ sont des zéros. On élimine de l'ensemble K la communication saturée k_{25}.

Itération 2. L'ensemble P' des points marqués comprend les points

$$p_1,\ p_2\,(1),\ p_6\,(2),\ p_8\,(6).$$

On passe au flot $X^{(2)}$ en modifiant $X^{(1)}$ le long de l'itinéraire obtenu k_{12}, k_{26}, k_{68}:

$$x_{12}^{(2)} = 2 + \theta,\quad x_{26}^{(2)} = \theta,\quad x_{68}^{(12)} = 0;\quad \theta = \min(1,\ 1,\ 2) = 1.$$

On rejette de l'ensemble K les communications k_{12} et k_{26}.

Itération 3. L'ensemble P' se compose des points

$$p_1,\ p_3\,(1),\ p_6\,(3),\ p_8\,(6).$$

Le passage au flot $X^{(3)}$ se ramène au calcul de

$$x_{13}^{(3)} = x_{36}^{(3)} = \theta;\quad x_{68} = 1 + \theta;\quad \theta = \min(4,\ 2,\ 1) = 1.$$

La communication k_{68} quitte l'ensemble K.

Itération 4.

$$P' = \{p_1,\ p_3\,(1),\ p_7\,(3),\ p_8\,(7)\}.$$

Le flot $X^{(4)}$ diffère de $X^{(3)}$ par les composantes

$$x_{13}^{(4)} = 1 + \theta,\quad x_{37}^{(4)} = x_{78}^{(4)} = \theta;\quad \theta = \min(3,\ 2,\ 5) = 2.$$

On fait sortir de l'ensemble K la communication k_{37}.

Itération 5.

$$P' = \{p_1,\ p_4\,(1),\ p_5\,(4),\ p_8\,(5)\}.$$

On est encore dans le cas b); on passe au flot $X^{(5)}$ en changeant les valeurs des composantes suivantes du flot $X^{(4)}$:

$$x_{14}^{(5)} = x_{45}^{(5)} = \theta,\quad x_{58}^{(5)} = 2 + \theta;\quad \theta = \min(5,\ 2,\ 4) = 2.$$

La communication exclue de l'ensemble K est k_{45}.

Itération 6. L'ensemble P' reçoit successivement les points

$$p_1,\ p_3\,(1),\ p_6\,(3),\ p_4\,(1).$$

L'extension ultérieure de l'ensemble des points marqués devient impossible. L'itération s'achève par le résultat 2.

La succession des calculs de la partie préalable de l'algorithme est traduite par la figure 1.11.

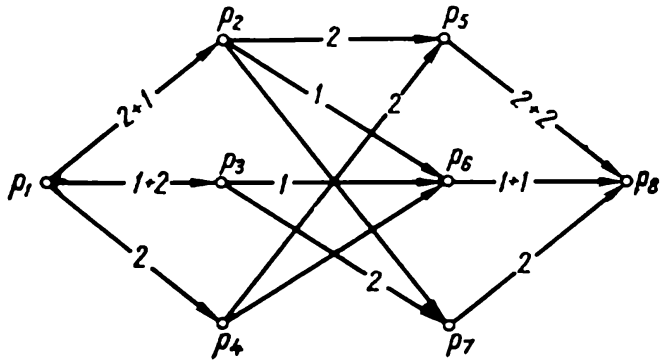

Fig. 1.11

On passe à la partie principale de l'algorithme.

I t é r a t i o n 7. La figure 1.12 représente le réseau de transport $(P, K^{(6)})$, où $K^{(6)}$ est l'ensemble des communications $X^{(6)}$-insaturées,

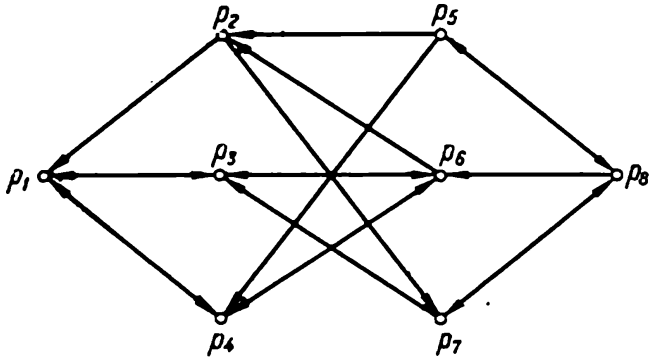

Fig. 1.12

$k_{ij} \in K'$. Ce réseau précisément rend commode la construction de l'ensemble P'.

Les points

$$p_1,\ p_3\ (1),\ p_6\ (3),\ p_2\ (6),\ p_7\ (2),\ p_8\ (7)$$

sont inclus successivement dans l'ensemble P'. L'itération se compose de cinq étapes dont la dernière conduit au résultat b). Il en résulte l'itinéraire entre p_1 et p_8 construit à partir des communications $X^{(6)}$-insaturées

$$k_{13},\ k_{36},\ k_{62},\ k_{27},\ k_{78}.$$

Le flot $X^{(7)}$ diffère du flot $X^{(6)}$ par les composantes

$$x_{13}^{(6)} = 3 + \theta, \quad x_{36}^{(6)} = 1 + \theta,$$
$$x_{26}^{(6)} = 1 - \theta, \quad x_{27} = \theta,$$
$$x_{78} = 2 + \theta; \quad \theta = \min\,(1,\ 1,\ 1,\ 1,\ 3) = 1.$$

Itération 8. Le réseau de transport $(P, K^{(7)})$, où $K^{(7)}$ est un système de communications $X^{(7)}$-insaturées de l'ensemble K', est représenté figure 1.13.

En se servant du réseau $(P, K^{(7)})$, on compose l'ensemble P' :

$$p_1,\ p_4\ (1),\ p_6\ (4),\ p_3\ (6).$$

La quatrième étape de l'itération s'achève par le cas c). Le marquage ultérieur des points devient impossible. Par conséquent, le

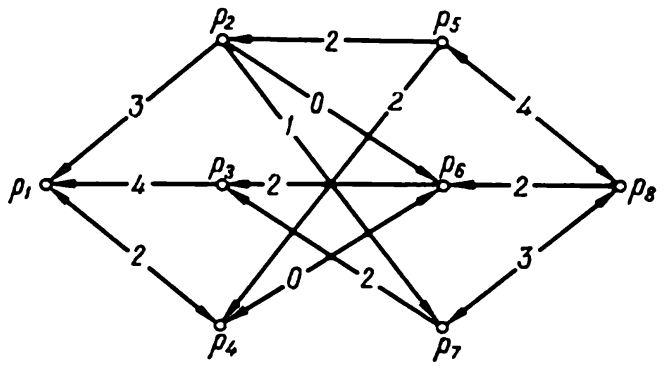

Fig. 1.13

flot $X^{(7)}$ est le flot maximal recherché. Les composantes du flot $X^{(7)}$ sont indiquées sur la figure 1.13, tous les chiffres se rapportant à $x_{ij}^{(7)}$ avec $i < j$. La valeur x_1^* du flot maximal est égale à 9.

On obtient en même temps la coupe minimale Γ^* du réseau, composée des communications

$$k_{12},\ k_{45},\ k_{68};\ k_{62},\ k_{37},$$
$$d\,(\Gamma^*) = 3 + 2 + 2 + 0 + 2 = 9 = x_1^*.$$

§ 4. Formulation en termes de réseaux et formulation matricielle des problèmes de transport

4.1. Ce paragraphe donne la formulation générale du problème de transport sur un réseau et met en lumière ses relations réciproques avec les problèmes du type T et T_d, étudiés en détail dans les chapitres 13 et 14 de [87].

Soit le réseau de transport quelconque (P, K). La fonction de fabrication et de consommation $q\,(p_i)$ est définie sur l'ensemble P des points de ce réseau ; la fonction $d\,(k_{ij})$ des capacités des communications du réseau et la fonction $c\,(k_{ij}) \geqslant 0$ du coût de transport sont définies sur l'ensemble K.

Au § 1.4 a été donnée la définition d'un programme de transports sur le réseau (P, K) compatible avec les fonctions $q\,(p_i)$, $p_i \in P$ et $d\,(k_{ij})$, $k_{ij} \in K$. Pour établir l'énoncé d'un problème de transport on peut soit partir de la notion de programme algébrique, soit utili-

ser l'écriture arithmétique du programme. Voici d'abord la position du problème associée aux programmes algébriques, c'est-à-dire aux fonctions $X = x(k_{ij})$, $k_{ij} \in K'$ qui satisfont aux conditions

$$x(p_i, P) = \sum_{k_{ij} \in E_i} x(k_{ij}) = q(p_i), \quad i = 1, 2, \ldots, N; \tag{4.1}$$

$$x(k_{ij}) \leqslant d(k_{ij}), \quad k_{ij} \in K'; \tag{4.2}$$

$$x(k_{ij}) = -x(k_{ji}). \tag{4.3}$$

Rappelons que E_i est l'ensemble des communications $k_{ij} \in K'$ issues du point p_i; K' est formé à partir de K par adjonction des communications $k_{ij} \notin K$ telles que $k_{ji} \in K$. Pour plus de commodité nous considérerons, comme d'habitude, que les fonctions X et D sont définies pour tout k_{ij}, en posant $x(k_{ij}) = d(k_{ij}) = 0$ si $k_{ij} \notin K'$.

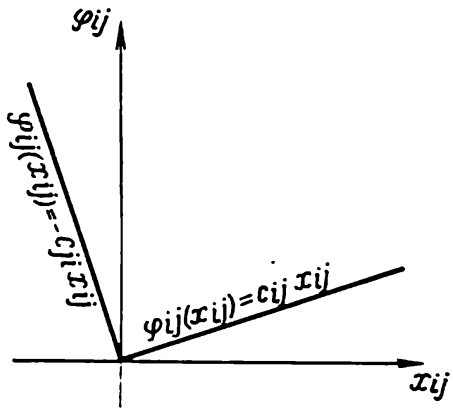

Fig. 1.14

Formons à l'aide des valeurs de la fonction $c(k_{ij})$ le système des fonctions $\varphi_{ij}(x_{ij})$ qui déterminent le coût d'un transport x_{ij} dans la communication k_{ij}:

$$\varphi_{ij}(x_{ij}) = \begin{cases} c_{ij}x_{ij} & \text{pour } x_{ij} \geqslant 0, \\ -c_{ji}x_{ij} & \text{pour } x_{ij} < 0. \end{cases}$$

Dans le cas général, la fonction $\varphi_{ij}(x_{ij})$ est une fonction linéaire par morceaux à convexité dirigée vers le bas et avec une seule brisure (fig. 1.14).

Si les points p_i et p_j sont reliés par une communication, les fonctions $\varphi_{ij}(x_{ij})$ et $\varphi_{ji}(x_{ji})$ sont linéaires sur les segments où elles sont définies.

Le problème de transport sur le réseau (P, K) avec la fonction de fabrication et de consommation $Q = q(p_i)$, $p_i \in P$, la fonction des capacités D et la fonction du coût de transport C consiste à construire un programme de transports $X = x(k_{ij})$, $k_{ij} \in K'$, compatible avec Q et D tel que le coût total

$$\Phi(X) = \frac{1}{2} \sum_{k_{ij} \in K'} \varphi_{ij}(x(k_{ij})) \tag{4.4}$$

soit minimal. Le problème (4.1)-(4.4) n'est pas un problème de programmation linéaire par suite de la non-linéarité de la fonction (4.4); toutefois, il se ramène aisément à un problème linéaire.

Si l'on utilise la notion de programme arithmétique, la fonction à minimiser devient linéaire, et le problème de transport se transfor-

me en problème linéaire. Pourtant les conditions (4.1) sont alors moins symétriques.

Le problème de transport sur le réseau (P, K) aux fonctions Q, D et C consiste en termes des programmes arithmétiques à minimiser la forme linéaire

$$L(X) = \sum_{k_{ij} \in K} c_{ij} x(k_{ij}) \tag{4.5}$$

sous les conditions

$$x(p_i, P) - x(P, p_i) = q(p_i), \quad i = 1, 2, \ldots, N, \tag{4.6}$$

$$0 \leqslant x(k_{ij}) \leqslant d(k_{ij}), \quad k_{ij} \in K. \tag{4.7}$$

Si $X = x(k_{ij})$, $k_{ij} \in K$, est un programme arithmétique,

$$Y = y(k_{ij}) = x(k_{ij}) - x(k_{ji}), \quad k_{ij} \in K'$$

est un programme algébrique.

D'une manière analogue, à chaque programme algébrique $Y = y(k_{ij})$, $k_{ij} \in K'$ correspond un programme arithmétique

$$X = x(k_{ij}) = \begin{cases} y(k_{ij}) \text{ pour } y(k_{ij}) \geqslant 0, \\ 0 \text{ pour } y(k_{ij}) < 0, \end{cases} \quad (k_{ij} \in K).$$

Il est facile de vérifier que les valeurs des fonctions (4.4) et (4.5) sur des programmes algébrique et arithmétique Y et X en correspondance mutuelle sont égales si $x(k_{ij}) \cdot x(k_{ji}) = 0$ pour $k_{ij} \in K$.

L'observation de cette condition rend biunivoque cette correspondance entre les programmes algébriques et arithmétiques; de plus, parmi les programmes arithmétiques qui minimisent la fonction (4.5) il y en a un qui vérifie cette condition. C'est pourquoi, si l'un des deux programmes liés par la relation biunivoque minimise la fonction qui lui correspond [(4.4) ou (4.5)], l'autre programme du couple considéré jouit également de cette propriété, ce qui signifie que les problèmes (4.1)-(4.4) et (4.5)-(4.7) sont équivalents. Pour abréger nous désignerons le problème (4.1)-(4.4) ou (4.5)-(4.7) par $T(q, d, c,)$.

Pour que le problème $T(q, d, c)$ ait une solution, il faut et il suffit qu'il existe au moins un programme compatible avec les fonctions $q(p_i)$ et $d(k_{ij})$.

La condition nécessaire pour la résolution du problème $T(q, d, c)$

$$\sum_{p_i \in P} q(p_i) = 0$$

a été obtenue au § 1.4. Dans ce qui suit, nous allons considérer que cette condition est remplie.

Constatons que les programmes compatibles avec les fonctions $q(p_i)$ et $d(k_{ij})$ s'appellent également programmes du problème $T(q, d, c)$.

Appelons selon l'usage programme optimal ou solution optimale du programme correspondant un programme qui minimise la fonction (4.4) ou (4.5).

Il est facile de ramener un problème $T(q, d, c)$ quelconque au cas où $q(p_i) = 0$ pour tout p_i sauf deux, l'entrée et la sortie.

Dans ce but, il suffit d'introduire deux nouveaux points du réseau p_0 et p_{N+1}, de mener à partir de p_0 des communications vers tous les points de fabrication, de relier chaque point de consommation par une communication avec p_{N+1}, de compléter la définition des $d(k_{ij})$ de la façon suivante :

$$d(k_{0j}) = q(p_j), \quad d(k_{i,\,N+1}) = -q(p_i),$$

où p_j sont les points de fabrication ;

p_i, les points de consommation,

et de poser enfin

$$c(k_{0j}) = c(k_{i,\,N+1}) = 0.$$

Soit

$$\overline{P} = \{P, p_0, p_{N+1}\}; \quad \overline{K} = \{K, k_{0j}, k_{i,\,N+1}\};$$

$\overline{D}$ et $\overline{C}$ sont les prolongements indiqués des fonctions D et C sur l'ensemble $\overline{K}$.

Posons

$$\overline{q}(p_i) = \begin{cases} \sum\limits_{q(p_\lambda)>0} q(p_\lambda); & i = 0, \\ \sum\limits_{q(p_\lambda)<0} q(p_\lambda), & i = N+1. \\ 0 \text{ pour} & 1 \leqslant i \leqslant N. \end{cases}$$

Le problème $T(q, d, c)$ est équivalent au problème $T(\overline{q}, \overline{d}, \overline{c})$. En effet, à tout programme X du problème $T(q, d, c)$ correspond le programme $\overline{X}$ du problème $T(\overline{q}, \overline{d}, \overline{c})$ tel que

$$\overline{x}_{ij} = \begin{cases} q(p_j) & \text{si } i = 0,\ q(p_j) > 0, \\ -q(p_i) & \text{si } j = N+1,\ q(p_i) < 0, \\ x_{ij} & \text{dans les autres cas,} \end{cases}$$

et, par ailleurs, les deux programmes sont associés à des coûts de transport identiques. D'autre part, pour tout programme $\overline{X}$ du problème $T(\overline{q}, \overline{d}, \overline{c})$,

$$\overline{x}_{0j} = d_{0j} = q(p_j), \ q(p_j) > 0,$$

$$\overline{x}_{i,\,N+1} = d_{i,\,N+1} = -q(p_i), \ q(p_i) < 0.$$

Par conséquent, le système de transports

$$x_{ij} = \overline{x}_{ij} \quad (i \neq 0,\ j \neq N+1)$$

est un programme X du problème $T(q, d, c)$; de plus, le passage au programme X ne change pas la valeur du coût de transport total.

La discussion des problèmes de transport sur un réseau (P, K) peut toujours s'effectuer en supposant que ce réseau est connexe. En effet, si le réseau de transport considéré est non connexe, on sait alors (cf. exercice 1) qu'il se décompose en un nombre fini t de sous-réseaux connexes maximaux

$$(P_\alpha, K_\alpha), \quad \alpha = 1, 2, \ldots, t.$$

Puisque les points du problème qui appartiennent aux différents ensembles P_α ne sont pas reliés par des communications de l'ensemble K, le problème $T(q, d, c)$ est équivalent aux t problèmes $T(q_\alpha, d_\alpha, c_\alpha)$, où $T(q_\alpha, d_\alpha, c_\alpha)$ est le problème de transport sur le réseau (P_α, K_α) et où les fonctions Q_α et D_α, C_α coïncident avec Q et D, C sur les ensembles P_α et K_α respectivement.

Dans ce qui suit, le réseau (P, K) du problème de transport $T(q, d, c)$ est considéré comme connexe.

4.2. Le problème de transport $T(q, d, c)$ écrit sous la forme (4.5)-(4.7) peut être considéré comme un problème de programmation linéaire à contraintes bilatérales. Le problème $T(q, d, c)$ compte s variables x_{ij} liées par N conditions d'égalité (4.6), où s est le nombre de communications $k_{ij} \in K$, N, le nombre de points $p_i \in P$.

Enonçons le critère d'optimalité d'un programme du problème $T(q, d, c)$, dont le bien-fondé se déduit directement du théorème 4.2 du chapitre 6 de [87].

Associons à la i-ème condition du système (4.6) ou, ce qui revient au même, au point p_i du réseau (P, K) le nombre u_i.

Soit

$$U = (-u_1, -u_2, \ldots, -u_N),$$

P_{ij} le vecteur des conditions du problème (4.5)-(4.7) associé à la variable x_{ij}.

La variable x_{ij} ne fait partie que de deux conditions du système (4.6), de la i-ème avec le facteur 1 et de la j-ième avec le facteur —1.

Autrement dit, la i-ème composante du vecteur P_{ij} est 1 et sa j-ième composante est —1, alors que les autres composantes sont nulles.

On en tire

$$(U, P_{ij}) = u_j - u_i. \tag{4.8}$$

Les raisonnements de la discussion du théorème 1.5 du chapitre 13 de [87] et l'égalité (4.8) permettent de tirer du théorème 4.2 du chapitre 6 de [87] la proposition suivante.

Théorème 4.1. [Critère d'optimalité d'un programme du problème (4.5)-(4.7)]. *Pour rendre optimal le programme* $X = x(k_{ij})$, $k_{ij} \in K$, *du problème* (4.5)-(4-7),

il faut et il suffit qu'il existe des nombres $u_1, u_2, \ldots, u_N$ *tels que*

$$u_j - u_i = c(k_{ij}) \text{ si } 0 < x(k_{ij}) < d(k_{ij}), \tag{4.9}$$

$$u_j - u_i \leqslant c(k_{ij}) \text{ si } x(k_{ij}) = 0,\ k_{ij} \in K, \tag{4.10}$$

$$u_j - u_i \geqslant c(k_{ij}) \text{ si } x(k_{ij}) = d(k_{ij}). \tag{4.11}$$

Les nombres u_i associés au programme optimal du problème (4.5)-(4.7) par les conditions (4.9)-(4.11) sont dits *potentiels* des points correspondants du réseau (P, K). L'ensemble des potentiels peut être considéré comme une certaine fonction $u(p_i)$ définie sur l'ensemble P.

Le théorème 4.1 permet d'obtenir aisément le critère d'optimalité du problème $T(q, d, c)$ écrit sous la forme (4.1)-(4.4).

Théorème 4.2. [Critère d'optimalité d'un programme du problème (4.1)-(4.4)]. *Pour rendre optimal le programme* $X = x(k_{ij})$, $k_{ij} \in K'$, *du problème* (4.1)-(4.4), *il faut et il suffit qu'il existe des nombres* $u_1, u_2, \ldots, u_N$ *tels que*

$$u_j - u_i = c(k_{ij}) \quad \text{si } 0 < x(k_{ij}) < d(k_{ij}), \tag{4.12}$$

$$u_j - u_i = -c(k_{ji}) \text{ si } -d(k_{ji}) < x(k_{ij}) < 0, \tag{4.13}$$

$$u_j - u_i \geqslant c(k_{ij}) \quad \text{si } x(k_{ij}) = d(k_{ij}), \tag{4.14}$$

$$u_j - u_i \leqslant -c(k_{ji}) \text{ si } x(k_{ij}) = -d(k_{ji}), \tag{4.15}$$

$$-c(k_{ji}) \leqslant u_j - u_i \leqslant c(k_{ij}) \quad \text{si } x(k_{ij}) = 0,\ k_{ij} \in K'. \tag{4.16}$$

Constatons que dans la condition (4.16) il est d'usage de considérer la valeur $c(k_{ij})$ comme infinie avec $k_{ij} \notin K$.

Nous laissons au lecteur le soin de démontrer les théorèmes 4.1 et 4.2 (cf. exercice 11).

Dans la discussion des problèmes du type $T(q, d, c)$ on recourt à la notion de programme de base du transport. Un programme arithmétique $X = x(k_{ij})$, $k_{ij} \in K$, est dit *de base* lorsqu'il est impossible de composer un itinéraire fermé à partir des communications $k_{ij} \in K$ telles que

$$0 < x(k_{ij}) < d(k_{ij}). \tag{4.17}$$

On peut vérifier (cf. exercice 12) que cette définition est équivalente à la définition courante d'un programme de base du problème $T(q, d, c)$ considéré comme un problème général de programmation linéaire à contraintes bilatérales.

Si le programme de transport X est écrit en termes algébriques, pour définir que c'est un programme de base on fait appel à la proposition suivante. Un programme algébrique $X = x(k_{ij})$, $k_{ij} \in K'$, est dit *de base* s'il est impossible de constituer un itinéraire orienté fermé à partir des communications $k_{ij} \in K'$ telles que

$$x(k_{ij}) \neq d_{ij},\quad 0,\quad -d_{ji}. \tag{4.18}$$

Les communications k_{ij} qui satisfont à la contrainte (4.17) dans le cas d'un programme arithmétique et sous la condition (4.18) s'appellent communications *principales* du programme X lorsque l'écriture du programme est algébrique. C'est pourquoi un programme de base peut être défini également comme un programme dont les communications principales rendent impossible la construction d'un itinéraire fermé (orienté dans le cas d'un programme algébrique).

Les raisonnements qui vont suivre montrent qu'à partir d'un programme de transport X disponible on peut construire un programme de base X' sans augmenter pour autant le coût de transport total.

Bornons-nous au cas d'un programme algébrique. Si le programme X n'est pas de base, il existe un itinéraire fermé orienté γ composé de ses communications principales. Supposons que $X(\theta)$ diffère de X par les transports

$$x(k_{ij}, \theta) = x(k_{ij}) + \theta \text{ avec } k_{ij} \in \gamma.$$

Puisque les valeurs des transports $x(k_{ij})$, $k_{ij} \in \gamma$, se situent à l'intérieur des intervalles de linéarité des fonctions $\varphi_{ij}(x_{ij})$, pour de petites valeurs absolues du paramètre θ

$$\Phi(X(\theta)) = \Phi(X) + \beta \cdot \theta, \text{ où } \beta = \text{const}.$$

Si $\beta < 0$ on augmente θ jusqu'à ce que l'un des transports de l'itinéraire γ devienne égal à sa valeur frontière (d_{ij}, 0 ou $-d_{ji}$).

Si $\beta \geqslant 0$, le paramètre θ doit être diminué *) en lui attribuant des valeurs négatives jusqu'à ce que l'un des transports de l'itinéraire γ devienne égal à sa valeur frontière. Dans les deux cas on aboutit à un nouveau programme dont le nombre de communications principales est plus petit et le coût de transport total n'est pas plus grand. Si le programme obtenu n'est pas de base, on reprend la procédure décrite en partant d'un nouvel itinéraire fermé.

Après quelques étapes on obtient le programme de base X' tel que

$$\Phi(X') \leqslant \Phi(X).$$

Le nombre d'étapes est fini du fait qu'après chacune d'elles le nombre de communications principales diminue.

On a ainsi démontré la proposition suivante:

Théorème 4.3. *Tout programme du problème $T(q, d, c)$ peut être transformé en son programme de base sans augmenter le coût de transport total.*

4.4. Les problèmes T et T_d examinés dans les chapitres 13 et 14 de [87] sont dits généralement formulations *matricielles* des problèmes de transport. Il en est ainsi parce qu'il est très commode d'écrire les données initiales et les programmes des problèmes T et T_d sous

*) Avec $\beta = 0$ on peut tomber dans le cas où, pour atteindre le but cherché, il faut augmenter le paramètre θ: $x(k_{ij}) < 0$, $c_{ij} = 0$ pour tout $k_{ij} \in \gamma$.

la forme de tableaux rectangulaires (matrices), et la procédure de leur résolution consiste le plus souvent à effectuer une série de transformations des matrices rectangulaires.

Les énoncés des problèmes (4.1)-(4.4) ou (4.5)-(4.7) étant liés aux réseaux, il est logique d'appeler ces problèmes formulations d'un problème de transport en termes *de réseau*.

Nous avons déjà dit que pour un réseau de transport simple privé de points de transit et dont les points de fabrication ne font qu'expédier le produit alors que les points de consommation ne font que de le recevoir, une formulation matricielle d'un problème de transport est équivalente à une formulation en termes de réseau. Nous allons montrer que la réciproque est vraie également: tout problème du type $T(q, d, c)$ peut se ramener à un problème de la forme $T_{\bar{d}}$.

Ainsi, soit $T(q, d, c)$ un problème de transport sur le réseau (P, K), écrit sous la forme (4.5)-(4-7). Posons

$$q = \sum_{q(p_i)>0} q(p_i) = \sum_{p_i \in P_+} q(p_i)$$

(q est la quantité totale de produit disponible aux points de fabrication).

Introduisons dans la discussion le problème $T_{\bar{d}}$ qui consiste à minimiser la forme linéaire

$$C(Z) = \sum_{i=1}^{N} \sum_{j=1}^{N} \bar{c}_{ij} z_{ij} \tag{4.19}$$

sous les conditions

$$\sum_{j=1}^{N} z_{ij} = q + q(p_i), \quad i = 1, 2, \ldots, N, \tag{4.20}$$

$$\sum_{i=1}^{N} z_{ij} = q, \qquad j = 1, 2, \ldots, N, \tag{4.21}$$

$$0 \leqslant z_{ij} \leqslant \bar{d}_{ij}, \quad i = 1, 2, \ldots, N; \quad j = 1, 2, \ldots, N, \tag{4.22}$$

où N est le nombre de points du réseau (P, K):

$$\bar{c}_{ij} = \begin{cases} c(k_{ij}) & \text{si } k_{ij} \in K, \\ 0 & \text{si } i = j, \\ \infty & \text{si } i \neq j \text{ et } k_{ij} \notin K, \end{cases}$$

$$\bar{d}_{ij} = \begin{cases} d(k_{ij}) & \text{si } k_{ij} \in K, \\ \infty & \text{si } k_{ij} \notin K. \end{cases}$$

1. Supposons que le problème (4.19)-(4.22) ait une solution et qu'en outre, la valeur minimale de la forme linéaire (4.19) soit finie. Soit $\| z_{ij} \|_N$ un programme optimal de ce problème. En vertu de la

définition des nombres $\bar{c}_{ij}$, le transport $z_{ij} = 0$ si $i \neq j$ et $k_{ij} \notin K$. Posons

$$x(k_{ij}) = z_{ij} \text{ pour } k_{ij} \in K. \tag{4.23}$$

Les conditions (4.20)-(4.21) peuvent s'écrire maintenant

$$\sum_{k_{ij} \in E'_i} x(k_{ij}) + z_{ii} = q + q(p_i),$$

$$\sum_{k_{ij} \in E''_j} x(k_{ij}) + z_{jj} = q.$$

E'_i (E''_j) est ici l'ensemble des communications issues de p_i (entrant dans p_j).

En retranchant de l'égalité supérieure pour $i = \lambda$ l'égalité inférieure pour $j = \lambda$, il vient

$$x(p_\lambda, P) - x(P, p_\lambda) = \sum_{k_{\lambda j} \in E'_\lambda} x(k_{\lambda j}) -$$

$$- \sum_{k_{i\lambda} \in E''_\lambda} x(k_{i\lambda}) = q(p_\lambda), \quad \lambda = 1, \ldots, N.$$

(4.22) entraîne

$$0 \leqslant x(k_{ij}) \leqslant d(k_{ij}).$$

Ainsi, la fonction $x(k_{ij})$ définie sur l'ensemble K par les égalités (4.23) est un programme du problème (4.5)-(4.7). Dans ces conditions les valeurs des formes linéaires (4.5) et (4.19) sont respectivement égales sur les programmes $X = x(k_{ij})$, $k_{ij} \in K$, et $Z = \| z_{ij} \|_N$. Montrons que le programme X est une solution du problème (4.5)-(4.7). Supposons qu'il existe un programme $X' = x'(k_{ij})$, $k_{ij} \in K$, associé au coût de transport total

$$L(X') < L(X) = C(Z). \tag{4.24}$$

Considérons que le réseau (P, K) n'a pas d'itinéraires fermés orientés constitués de communications k_{ij} tels que $x'(k_{ij}) > 0$. Cette hypothèse ne réduit pas la généralité, puisque dans le cas contraire, en diminuant la valeur des transports le long des itinéraires fermés on obtient un nouveau programme $x''(k_{ij})$, $k_{ij} \in K$, qui jouit de la propriété concernée, et de plus

$$L(X'') \leqslant L(X')$$

(cf. démonstration du théorème 4.3).

Au lieu de X', on peut prendre le programme X'' sans compromettre pour autant la condition (4.24).

Posons

$$z'_{ij} = \begin{cases} x'(k_{ij}) & \text{si } k_{ij} \in K, \\ 0 & \text{si } i \neq j \text{ et } k_{ij} \notin K, \\ q - \sum\limits_{k_{ij} \in E''_j} x'(k_{ij}) & \text{avec } i = j. \end{cases} \tag{4.25}$$

D'après (4.25),

$$\sum_{i=1}^{N} z'_{ij} = \sum_{k_{ij} \in E'_j} x'(k_{ij}) - \sum_{k_{ij} \in E''_j} x'(k_{ij}) + q = q, \quad i = 1, 2, \ldots N.$$

Ensuite, les conditions (4.6) et la représentation (4.25) amènent

$$\sum_{j=1}^{N} z'_{ij} = \sum_{k_{ij} \in E'_i} x'(k_{ij}) - \sum_{k_{ji} \in E''_i} x'(k_{ji}) + q = q + q_i, \quad i = 1, 2, \ldots, N.$$

Comme les transports $x'(k_{ij})$ vérifient les conditions (4.7), les nombres z'_{ij} sont justifiés par les contraintes (4.22) si et seulement si

$$q - \sum_{k_{ij} \in E''_j} x'(k_{ij}) \geqslant 0. \tag{4.26}$$

Montrons que l'inégalité (4.26) est vraie pour n'importe quelle valeur de j.

Soient Γ_1 l'ensemble des points $p_i \in P$ qui acheminent les transports positifs x'_{ij} vers p_j, Γ_2 l'ensemble des points qui d'après le programme X' assurent l'approvisionnement des points de l'ensemble Γ_1, etc. Les ensembles Γ_λ, $\lambda = 1, 2, \ldots, t$, sont disjoints, car s'il n'en est pas ainsi on pourrait trouver sur le réseau (P, K) un itinéraire fermé orienté composé de communications à transports positifs.

Le nombre t des ensembles Γ_λ ne dépasse pas $N - 1$. Compte tenu des conditions (4.6) et de la définition des ensembles Γ_λ, on a

$$\sum_{k_{ij} \in E''_j} x'(k_{ij}) = x'(\Gamma_1, p_j) \leqslant x'(\Gamma_1, P) = x'(P, \Gamma_1) + q(\Gamma_1) =$$
$$= x'(\Gamma_2, \Gamma_1) + q(\Gamma_1).$$

En reprenant les mêmes raisonnements, on aboutit aux inégalités

$$x'(\Gamma_2, \Gamma_1) \leqslant x'(\Gamma_3, \Gamma_2) + q(\Gamma_2),$$
$$\cdots\cdots\cdots\cdots\cdots\cdots$$
$$\cdots\cdots\cdots\cdots\cdots\cdots$$
$$x'(\Gamma_{t-1}, \Gamma_{t-2}) \leqslant x'(\Gamma_t, \Gamma_{t-1}) + q(\Gamma_{t-1}).$$

En additionnant les $t - 1$ inégalités obtenues, on a

$$x'(\Gamma_1, p_j) \leqslant x'(\Gamma_t, \Gamma_{t-1}) + \sum_{\lambda=1}^{t-1} q(\Gamma_\lambda). \tag{4.27}$$

D'après le programme X', les points de l'ensemble Γ_t ne reçoivent pas de produit en provenance des autres points. On a donc

$$x'(\Gamma_t, \Gamma_{t-1}) \leqslant x'(\Gamma_t, P) \leqslant q(\Gamma_t) + x'(P, \Gamma_t) = q(\Gamma_t). \tag{4.28}$$

Les inégalités (4.27) et (4.28) conduisent à la relation

$$x'(\Gamma_1, p_j) \leqslant \sum_{\lambda=1}^{t} q(\Gamma_\lambda) \leqslant q,$$

équivalente à (4.26).

Ainsi le système de nombres $Z' = \| z'_{ij} \|_N$ défini par la formule (4.25) est un programme du problème (4.19)-(4.22). De plus,

$$L(X') = C(Z').$$

La comparaison de cette dernière égalité avec l'inégalité (4.24) conduit à la relation

$$C(Z') < C(Z),$$

qui contredit l'optimalité du programme Z.

Par conséquent (4.24) est impossible, ce qui signifie que le système de transport X associé au programme Z par les égalités (4.23) constitue réellement la solution du problème (4.5)-(4.7).

2. Considérons l'autre éventualité à laquelle peut aboutir la discussion du problème (4.19)-(4.22): les conditions (4.20)-(4-22) sont contradictoires, ou bien la valeur minimale de la forme linéaire (4.19) est infinie. Ce cas est équivalent à l'incompatibilité du système composé des conditions (4.20)-(4.22) et des contraintes

$$z_{ij} = 0 \text{ si } i \neq j \text{ et } k_{ij} \notin K. \tag{4.29}$$

Montrons que l'incompatibilité du système (4.20)-(4.22), (4.29) n'est possible que si le problème (4.5)-(4.7) ne possède aucun programme.

En effet, soit X' un programme du problème (4.5)-(4.7). Nous avons déjà vu qu'on ne réduit pas la généralité en adoptant l'hypothèse sur l'absence d'itinéraires fermés orientés constitués de communications du réseau (P, K) le long desquelles on transporte une quantité positive de produit. Par suite, d'après ce qui a été démontré, la matrice $Z' = \| z'_{ij} \|_N$, définie par la formule (4.25), est un programme du problème (4.19)-(4.22). Par définition des nombres z'_{ij}, le programme Z' satisfait également aux conditions (4.29).

Ainsi, l'hypothèse de l'existence d'un programme du problème (4.5)-(4.7) rend possible la résolution du système (4.20)-(4.22), (4.29), ce qui démontre notre proposition. Résumons les conclusions obtenues sous la forme d'un théorème.

Théorème 4.4. *La discussion d'un problème de transport arbitraire sur un réseau contenant N points peut être ramenée à la résolution d'un problème de transport du type T_d à N points de fabrication et N points de consommation.*

4.5. Dans certains cas, la taille d'un problème de transport équivalent énoncé en formulation matricielle peut être nettement réduite.

Considérons le problème de transport $T(q, d, c) = T(q, c)$ sur un réseau (P, K) dont toutes les communications ont une capacité illimitée, c'est-à-dire $d(k_{ij}) = \infty$ pour tout $k_{ij} \in K$.

Soient P_+ l'ensemble des points de fabrication du problème, P_- celui de ses points de consommation. Pour la suite il est commode de donner une notation différente aux points de fabrication et aux points de consommation. Posons

$$P_+ = \{p'_1, p'_2, \ldots, p'_m\}; \quad P_- = \{p''_1, p''_2, \ldots, p''_n\}.$$

Composons pour chaque couple de points $p'_\lambda \in P_+$ et $p''_\mu \in P_-$ l'itinéraire orienté $l_{\lambda\mu}$ de p'_λ à p''_μ le moins cher composé de communications $k_{ij} \in K$. Désignons la valeur $c(l_{\lambda\mu})$ du coût de transport de l'itinéraire $l_{\lambda\mu}$ par $r_{\lambda\mu}$. S'il n'existe aucun itinéraire de p'_λ à p''_μ, la grandeur $r_{\lambda\mu}$ est posée infinie.

Les itinéraires optimaux $l_{\lambda\mu}$ et les nombres $r_{\lambda\mu}$ peuvent être recherchés, par exemple, par la méthode de Bellman-Shimbell décrite au § 2.4.

Introduisons dans la discussion le problème de transport qui consiste à minimiser la forme linéaire

$$R(Z) = \sum_{\lambda=1}^{m} \sum_{\mu=1}^{n} r_{\lambda\mu} z_{\lambda\mu} \tag{4.30}$$

sous les conditions

$$\sum_{\mu=1}^{n} z_{\lambda\mu} = q(p'_\lambda), \qquad \lambda = 1, 2, \ldots, m, \tag{4.31}$$

$$\sum_{\lambda=1}^{m} z_{\lambda\mu} = -q(p''_\mu), \qquad \mu = 1, 2, \ldots, n, \tag{4.32}$$

$$z_{\lambda\mu} \geqslant 0. \tag{4.33}$$

La solution du problème (4.30)-(4.33) définit l'ensemble optimal de livraisons des points de fabrication p'_λ aux points de consommation p''_μ sous la condition de n'utiliser que les itinéraires les moins chers du réseau (P, K). Les raisonnements qui suivent établissent l'équivalence des problèmes (4.5)-(4.7) et (4.30)-(4.33).

Le problème (4.30)-(4.33) a une solution puisque

$$\sum_{p'_\lambda \in P_+} q(p'_\lambda) = \sum_{p''_\mu \in P_-} \{-q(p''_\mu)\}.$$

Soit $Z = \| z_{ij} \|_{m.\,n}$ le programme optimal du problème.

1. Supposons d'abord que $R(Z) < \infty$. Dans ce cas

$$z_{\lambda\mu} = 0 \text{ si } c(l_{\lambda\mu}) = r_{\lambda\mu} = \infty. \tag{4.34}$$

Déterminons le système de transport $x_{\lambda\mu}(k_{ij})$, $k_{ij} \in K$, pour chaque couple de points $p'_\lambda \in P_+$ et $p''_\mu \in P_-$ tels que $r_{\lambda\mu} < \infty$, en

posant

$$x_{\lambda\mu}(k_{ij}) = \begin{cases} z_{\lambda\mu} & \text{si } k_{ij} \in l_{\lambda\mu}, \\ 0 & \text{dans les autres cas.} \end{cases} \tag{4.35}$$

Soit

$$X = x(k_{ij}) = \sum_{\lambda=1}^{m} \sum_{\mu=1}^{n} x_{\lambda\mu}(k_{ij}), \qquad k_{ij} \in K. \tag{4.36}$$

Les conditions (4.31)-(4.33) et (4.35) vérifiées par les composantes du programme Z entraînent que X est un programme du problème (4.5)-(4.7).

Par ailleurs,

$$C(X) = R(Z). \tag{4.37}$$

Montrons que le programme X est optimal. A cette fin il suffit de prouver que quel que soit le programme X' du problème $T(q, c)$, il existe un programme Z' du problème (4.30)-(4.33) associé à X' par l'inégalité

$$R(Z') \leqslant C(X'). \tag{4.38}$$

En effet, l'optimalité du programme Z et les conditions (4.37), (4.38) conduisent à la relation

$$C(X) = R(Z) \leqslant R(Z') \leqslant C(X'),$$

vraie pour tout programme X' du problème (4.5)-(4.7).

Ainsi, soit X' le programme arbitraire du problème (4.5)-(4.7). Sans restreindre la généralité considérons que X' est un programme de base, étant donné que s'il n'en est pas ainsi le théorème (4.3) rend possible le passage au programme de base X'' tel que

$$C(X'') \leqslant C(X'),$$

auquel on applique les considérations qui suivent.

Pour chaque couple de points p'_λ et p''_μ il existe au plus un itinéraire orienté de p'_λ à p''_μ constitué de communications principales du programme X'. Choisissons un couple quelconque de points p'_λ et p''_μ pour lequel l'itinéraire indiqué l_1 existe et posons

$$Y_1 = y_1(k_{ij}) = \begin{cases} \varepsilon_1 & \text{si } k_{ij} \in l_1, \\ 0 & \text{dans les autres cas,} \end{cases} \tag{4.39}$$

où

$$\varepsilon_1 = \min_{k_{ij} \in l_1} \{x'(k_{ij}),\ q(p'_\lambda),\ -q(p''_\mu)\}. \tag{4.40}$$

Considérons ensuite le système de transport

$$X' - Y_1,$$

qui est le programme du problème $T(q_1, c)$ où

$$q_1(p_i) = \begin{cases} q(p_i) & \text{avec } p_i \neq p'_\lambda,\ p''_\mu, \\ q(p_i) + \varepsilon_1 & \text{avec } p_i = p''_\mu, \\ q(p_i) - \varepsilon_1 & \text{avec } p_i = p'_\lambda, \end{cases}$$

choisissons un couple de points p'_{λ_1}, p''_{μ_1} ($q_1(p'_{\lambda_1}) > 0$, $q_1(p''_{\mu_1}) < 0$) pouvant être reliés par un itinéraire orienté l_2 composé de communications principales du programme $X' - Y_1$, et construisons un système de transport Y_2 en utilisant les formules (4.39), (4.40), dans lesquelles l'indice 1 est remplacé par 2.

Poursuivons cette procédure tant que cela est possible. Puisque chaque étape réduit le nombre de communications principales ou bien diminue le nombre de points de fabrication ou de consommation, on obtient en un nombre fini d'étapes un programme de base de transport

$$X_0 = X' - \sum_{i=1}^{t} Y_i$$

du problème $T(q_t, c)$, qui jouit de la propriété suivante. Si $q_t(p'_\lambda) > 0$ et $q_t(p''_\mu) < 0$, l'itinéraire orienté de p'_λ à p''_μ constitué de communications principales du programme X_0 n'existe pas.

On vérifie sans peine que $X_0 = 0$. En effet, supposons que pour une certaine communication k_{ij} le transport $x_0(k_{ij}) > 0$. Avec $q = q_t$ les conditions (4.6), vérifiées par le programme de base X_0, conduisent à deux couples d'alternatives:

a) $q_t(p_j) < 0$, ou bien pour un certain j_1 le transport $x_0(k_{jj_1}) > 0$;

b) $q_t(p_i) > 0$, ou bien $x_0(k_{i_1 i}) > 0$ pour un certain i_1.

En appliquant les mêmes raisonnements aux communications $k_{i_1 i}$, k_{jj_1}, etc., et en tenant compte du fait que X_0 est un programme de base, on arrive à la conclusion qu'il existe un itinéraire orienté composé de communications principales de X_0 et qui relie un couple de points de fabrication et de consommation du problème $T(q_t, c)$. Cette contradiction ne peut être éliminée qu'avec $X_0 = 0$.

Ainsi,

$$X' = \sum_{i=1}^{t} Y_i. \tag{4.41}$$

Déterminons de la façon suivante les collections de nombres $z'_{\lambda\mu}$ et $r'_{\lambda\mu}$ pour $1 \leqslant \lambda \leqslant m$, $1 \leqslant \mu \leqslant n$. Si les points p'_λ et p''_μ, reliés par l'itinéraire l_s, ont participé à la construction de Y_s,

$$z'_{\lambda\mu} = \varepsilon_s; \quad r'_{\lambda\mu} = c(l_s);$$

si par contre ils n'ont pris part à la construction d'aucun Y_i,

$$z'_{\lambda\mu} = 0, \quad r'_{\lambda\mu} = \infty.$$

Suivant l'égalité (4.41), la matrice $Z' = \| z'_{\lambda\mu} \|_{m,\, n}$ est un programme du problème (4.30)-(4.33), et de plus

$$C(X') = \sum_{\lambda=1}^{m} \sum_{\mu=1}^{n} r'_{\lambda\mu} z'_{\lambda\mu}.$$

Etant donné que $r'_{\lambda\mu}$ est la valeur du coût de transport d'un certain itinéraire de p'_λ à p''_μ, il vient

$$r_{\lambda\mu} \leqslant r'_{\lambda\mu}.$$

Par conséquent,

$$R(Z') = \sum_{\lambda=1}^{m} \sum_{\mu=1}^{n} r_{\lambda\mu} z'_{\lambda\mu} \leqslant \sum_{\lambda=1}^{m} \sum_{\mu=1}^{n} r'_{\lambda\mu} z'_{\lambda\mu} = C(X').$$

Le programme Z' recherché est ainsi construit, ce qui montre que le programme Z est optimal.

2. Supposons maintenant que $R(Z) = \infty$. Dans ce cas les conditions du problème $T(q, c)$ sont contradictoires. En effet, si l'on raisonne par l'absurde, c'est-à-dire si l'on suppose l'existence d'un certain programme X' du problème $T(q, c)$, on peut construire un programme Z' du problème (4.30)-(4.33) tel que

$$R(Z') \leqslant C(X') < \infty.$$

La contradiction obtenue montre que les conditions du problème sont incompatibles.

Nous avons donc démontré la proposition suivante.

Théorème 4.5. *La discussion d'un problème quelconque du type $T(q, c)$ à m points de fabrication et n points de consommation se ramène à la résolution d'un problème du type T au même nombre de points.*

La matrice R du coût de transport du problème équivalent se détermine en construisant un système d'itinéraires optimaux entre les points de fabrication et les points de consommation du problème initial.

4.6. Le mode d'établissement d'une formulation matricielle équivalente simple de conception qui a été exposé au paragraphe précédent, n'est applicable qu'aux problèmes du type $T(q, c)$. Toutefois, comme nous allons le voir, tout problème $T(q, d, c)$ peut être ramené à un problème de transport sur un réseau dont les communications ont une capacité illimitée. Considérons une communication quelconque k_{ij} du réseau (P, K) ayant une capacité limitée d_{ij}.

Transformons le réseau de transport (P, K) en un réseau (P_1, K_1) de la manière suivante. Adjoignons les points s' et s'' et les communications de p_i à s', de s'' à p_j et de s'' à s' ; éliminons la communication k_{ij} (fig. 1.15). Complétons ensuite la définition des fonctions Q,

D et C en posant

$$q(s') = -d_{ij}, \quad q(s'') = d_{ij},$$
$$c(p_i, s') = c(s'', s') = 0, \quad c(s'', p_j) = c_{ij},$$
$$d(p_i, s') = d(s'', s') = d(s'', p_j) = \infty.$$

Soit $T_1(q, d, c)$ le problème de transport sur le réseau (P_1, K_1) associé aux fonctions Q, D et C dont la définition vient d'être complétée. Nous allons montrer que les problèmes $T(q, d, c)$ et

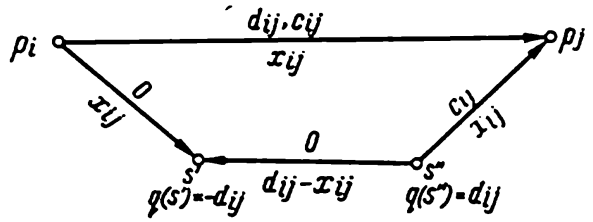

Fig. 1.15

$T_1(q, d, c)$ sont équivalents. En effet, considérons un programme quelconque $X = x(k_{ij})$, $k_{ij} \in K$, du problème $T(q, d, c)$. Si l'on pose

$$x(p_i, s') = x(s'', p_j) = x(k_{ij}), \quad x(s'', s') = d_{ij} - x_{ij},$$

la fonction $X_1 = x(k_{\lambda\mu})$, $k_{\lambda\mu} \in K_1$, est un programme du problème $T_1(q, d, c)$, et de plus

$$C(X) = C(X_1). \tag{4.42}$$

Soit maintenant $X_1 = x(k_{ij})$, $k_{ij} \in K_1$, le programme arbitraire du problème $T_1(q, d, c)$. Il est clair que

$$x(p_i, s') = x(s'', p_j) = x_{ij}, \quad 0 \leqslant x_{ij} \leqslant d_{ij}.$$

Par conséquent, si l'on considère que $x(k_{ij}) = x_{ij}$, on obtient que $X = x(k_{\lambda\mu})$, $k_{\lambda\mu} \in K$, est un programme du problème $T(q, d, c)$; de plus, on a de nouveau l'égalité (4.42).

Ainsi, une communication à capacité limitée peut être éliminée en introduisant dans le réseau de transport deux nouveaux points. En appliquant le procédé décrit à tout $k_{ij} \in K$ tel que $d_{ij} < \infty$, on obtient un problème équivalent $T(q, c)$ à $m + \gamma$ points de fabrication et $n + \gamma$ points de consommation, m et n désignant ici respectivement les nombres de points de fabrication et de consommation du problème initial, γ étant égal au nombre de communications $k_{ij} \in K$ à capacité limitée.

A titre de conclusion de ce paragraphe, il convient de noter qu'il est avantageux de ramener les problèmes de transport sur réseau du type $T(q, c)$ à une forme matricielle en se guidant par les recommandations du § 4.5. Lorsque le problème initial compte un grand

nombre de points de transit, cette transformation réduit nettement la procédure.

Si la fonction $d(k_{ij})$ prend une valeur finie sur un nombre relativement petit de communications $k_{ij} \in K$ et si le nombre de points de transit est important, il vaut mieux d'abord explorer un problème équivalent du type $T(q, c)$ pour le ramener ensuite à une forme matricielle par la méthode du § 4.5. Il est peu probable que la réduction à la forme matricielle des problèmes du type $T(q, d, c)$ de grande taille dans lesquels de nombreuses communications ont une capacité limitée puisse apporter un avantage quelconque. Mieux vaut résoudre ces problèmes en utilisant l'une des méthodes d'exploration des problèmes de transport sur réseaux, dont la description est donnée dans les deux paragraphes suivants de ce chapitre. Dans le cas du calcul manuel, les problèmes de transport donnés en termes de réseau présentent un avantage par rapport aux formulations matricielles, dû au caractère concret de leur énoncé. Souvent, la représentation graphique du problème permet de construire une variante valable sans recourir à des algorithmes formels quelconques.

§ 5. Méthode des potentiels

5.1. Considérons un problème quelconque du type $T(q, d, c)$, associé au réseau de transport (P, K). Supposons que ce problème soit énoncé en termes de programmes algébriques, c'est-à-dire qu'il ait la forme (4.1)-(4.4). La méthode de résolution du problème $T(q, d, c)$ décrite dans ce paragraphe est une extension naturelle de la méthode des potentiels appliquée aux problèmes du type T_d (cf. le § 3 du chapitre 14 de [87]). C'est pourquoi nous lui gardons la même appellation. Avant d'aborder l'exposé de cette méthode, précisons la notion de programme de base non dégénéré d'un problème de transport sur un réseau.

Soit X un programme de base du problème $T(q, d, c)$. Associons en un ensemble K_X les communications principales de ce programme. Nous dirons que le programme X est *non dégénéré* si le réseau (P, K_X) est connexe, c'est-à-dire si deux points quelconques de l'ensemble P peuvent être reliés par un itinéraire composé de communications $k_{ij} \in K_X$. Le problème $T(q, d, c)$ dont tous les programmes de base jouissent de la propriété de non-dégénérescence est dit *non dégénéré*. Constatons que d'après la définition d'une communication principale du programme algébrique, les points p_i et p_j ne sont pas reliés par des communications de K_X, ou bien sont réunis par deux communications opposées de cet ensemble. L'ensemble K_X du programme de base non dégénéré X peut être donc considéré comme connexe orienté. Montrons que cet ensemble compte précisément $(N - 1)$ couples de communications opposées, où N est le nombre total de points du problème.

Les points du réseau de transport reliés au moins par une communication sont dits *voisins*. Introduisons les notations:

P_1, l'ensemble des points du réseau (P, K_X) voisins du point p_1;

P_i $(i = 2, 3, \ldots)$, l'ensemble des points du réseau (P, K_X) voisins de l'un des points de l'ensemble P_{i-1}.

Les ensembles P_i, $i = 1, 2, \ldots$, n'ont pas de points communs, puisqu'en raisonnant par l'absurde on aboutit à l'existence d'un itinéraire fermé composé de communications de l'ensemble K_X, incompatible avec le programme de base X. Il en résulte notamment l'existence d'un nombre t depuis lequel (avec $i > t$) les ensembles P_i ne contiennent aucun programme.

Le réseau (P, K_X) étant connexe, on en déduit que la réunion des ensembles P_i, $i = 0, 1, \ldots, t$, coïncide avec l'ensemble P (ici $P_0 = \{p_1\}$).

Faisons correspondre à chaque point $p_\lambda \in P_i$ un couple de communications opposées $k_{\lambda\mu}$ et $k_{\mu\lambda}$ de K_X, où μ est le numéro du point $p_\mu \in P_{i-1}$ voisin de p_λ $(i = 1, 2, \ldots, t)$. Désignons l'ensemble de toutes ces communications par $\overline{K}_X$.

Les égalités démontrées

$$P = \bigcup_{0 \leqslant i \leqslant s} P_i \quad P_{i_1} \cap P_{i_2} = 0 \quad \text{pour } i_1 \neq i_2, \tag{5.1}$$

font que $\overline{K}_X$ se compose de $N - 1$ couples de communications opposées. C'est pourquoi la démonstration de notre proposition se ramène à vérifier l'égalité

$$\overline{K}_X = K_X. \tag{5.2}$$

Soit k_{ij} la communication quelconque de l'ensemble K_X. D'après la définition de l'ensemble $\overline{K}_X$ et la première égalité (5.1), il existe un itinéraire l constitué de communications de $\overline{K}_X$ qui relie les points p_i et p_j. Si l'on avait $k_{ij} \notin \overline{K}_X$, les communications de l'itinéraire l et la communication $k_{ij} \notin l$ constitueraient un itinéraire fermé, ce qui contredit le fait que X est un programme de base.

Par conséquent, $k_{ij} \in \overline{K}_X$. L'égalité (5.2) est démontrée. Donc K_X est composé de $N - 1$ couples de communications opposées, et de plus, pour tout point de l'ensemble P_i il existe un seul point voisin de l'ensemble P_{i-1}. Les ensembles P_i peuvent être utilisés pour la recherche d'un itinéraire orienté entre p_1 et un point arbitraire $p_\mu \in P$. En effet, soit $p_\mu \in P_s$. Dégageons dans chaque ensemble P_α pour $\alpha \leqslant s$ un point p_{i_α} en se guidant sur la règle:

$$p_{i_s} = p_\mu; \quad p_{i_t} \text{ et } p_{i_{t+1}} \text{ sont des points voisins.}$$

D'après ce qui vient d'être démontré, le point p_{i_t} est défini univoquement par le point $p_{i_{t+1}}$ et $k_{i_t i_{t+1}} \in K_X$.

L'itinéraire qui passe par les points

$$p_1 = p_{i_0},\ p_{i_1},\ \ldots,\ p_{i_{s-1}},\ p_{i_s} = p_\mu$$

est l'itinéraire recherché.

5.2. La procédure de résolution du problème $T(q, d, c)$ par la méthode des potentiels se compose d'un nombre fini d'itérations. Chaque itération compte deux étapes, dont la première consiste à justifier l'optimalité du programme de base déjà obtenu. Si le programme disponible ne satisfait pas aux conditions d'optimalité, la deuxième étape permet de construire un programme de base nouveau. La méthode est exposée sous l'hypothèse que le problème $T(q, d, c)$ est non dégénéré. Les remarques nécessaires pour traiter le cas général seront faites dans ce qui suit.

Examinons de plus près une itération isolée de cette méthode. Supposons que les itérations précédentes aient abouti au programme de base $X = x(k_{ij})$, $k_{ij} \in K'$.

E t a p e 1. On choisit un point quelconque de l'ensemble P, p_1 par exemple, pour former en partant de ce point un système de potentiels préalables $u_i = u(p_i)$, $p_i \in P$, du programme X, c'est-à-dire un système de nombres qui vérifient les équations

$$u(p_j) = u(p_i) =$$

$$= \begin{cases} c_{ij} = c(p_i, p_j) = c(k_{ij}) & \text{pour } x(k_{ij}) > 0,\ k_{ij} \in K_X, \\ -c_{ji} = -c(p_j, p_i) = -c(k_{ji}) & \text{pour } x(k_{ij}) < 0,\ k_{ij} \in K_X. \end{cases} \tag{5.3}$$

A cette fin on compose les ensembles $P_0 = \{p_1\}$, P_1, P_2, ..., P_t examinés au § 5.1.

Les potentiels préalables $u(p_i)$ sont calculés successivement pour les points appartenant à P_0, P_1, P_2, etc., d'après la règle

$$u(p_1) = 0,\ u(p'') - u(p') = \begin{cases} c(p', p'') & \text{pour } x(p', p'') > 0, \\ -c(p'', p') & \text{pour } x(p', p'') < 0, \end{cases}$$

si $p'' \in P_i$ et $p' \in P_{i-1}$; de plus, p' et p'' sont des points voisins du réseau (P, K_X).

Les propriétés des ensembles P_i exposées au § 5.1 conduisent à la conclusion que la règle considérée rend possible le calcul univoque des valeurs $u(p_i)$ pour tout $p_i \in P$ vérifiant le système d'égalités (5.3).

On établit ensuite pour chaque communication non principale $k_{ij} \in K'$ la différence des potentiels $u(p_j) - u(p_i)$. Si les conditions

$$\begin{aligned} -c_{ji} \leqslant u(p_j) - u(p_i) \leqslant c_{ij} \quad & \text{avec } x(k_{ij}) = 0, \\ u(p_j) - u(p_i) \geqslant c_{ij} \quad & \text{avec } x(k_{ij}) = d_{ij}, \\ u(p_j) - u(p_i) \leqslant -c_{ji} \quad & \text{avec } x(k_{ij}) = -d_{ji} \end{aligned} \tag{5.4}$$

sont vérifiées, d'après le critère d'optimalité (théorème 4.2), le programme X est une solution du problème. Mais si l'une au moins des conditions (5.4) est compromise, on passe à l'étape 2.

Les inégalités (5.4) sont soit observées simultanément par un couple de communications opposées k_{ij} et k_{ji} soit ne le sont pas par toutes ces communications. C'est pourquoi avec p_i et p_j fixes les restrictions (5.4) sont vérifiées une seule fois (pour k_{ij} ou pour k_{ji}).

E t a p e 2. On considère les couples de points du réseau pour lesquels les conditions (5.4) sont compromises. On choisit parmi eux un certain couple p_λ, p_μ. Le critère du choix peut être fourni, par exemple, par la grandeur de la variable d'écart de l'inégalité compromise du système (5.4). La marche ultérieure de l'étape dépend de celle des conditions (5.4) qui n'est pas observée avec $i = \lambda$ et $j = \mu$. Considérons chacune de ces possibilités.

a) L'élément compromis est le second membre de la première inégalité du système (5.4):

$$u(p_\mu) - u(p_\lambda) > c_{\lambda\mu},\ x(k_{\lambda\mu}) = 0.$$

Il est évident que ce n'est possible que lorsque $c_{\lambda\mu} < \infty$, c'est-à-dire avec $k_{\lambda\mu} \in K$.

On construit un itinéraire orienté l de p_μ à p_λ en utilisant les communications $k_{ij} \in K_X$. Cette opération peut être réalisée à l'aide de la formation successive des ensembles P_i de la manière décrite au § 5.1, en prenant p_μ comme point initial. La formation des ensembles P_i se poursuit tant que le point p_λ tombe dans l'ensemble P_s pour un certain s. On trouve ensuite le point $p_{\lambda_{s-1}}$ voisin de p_λ (au sens du réseau (P, K_X)) et situé dans P_{s-1}. On passe de façon analogue de $p_{\lambda_{s-1}}$ à $p_{\lambda_{s-2}} \in P_{s-2}$, etc. On obtient finalement un itinéraire orienté l de la forme

$$k_{\mu\lambda_1},\quad k_{\lambda_1\lambda_2},\ \ldots,\quad k_{\lambda_{s-2}\lambda_{s-1}},\quad k_{\lambda_{s-1}\lambda}, \tag{5.5}$$

nécessaire à la transformation suivante du programme X.

Le nouveau programme X' s'obtient à partir de X en augmentant les composantes $x(k_{ij})$ le long de l'itinéraire fermé γ constitué par l'itinéraire (5.5) et la communication $k_{\lambda\mu}$, ce que traduit la relation

$$x'(k_{ij}) = \begin{cases} x(k_{ij}) + \theta & \text{avec } k_{ij} \in \gamma, \\ x(k_{ij}) - \theta & \text{avec } k_{ij} \in \gamma, \\ x(k_{ij}) & \text{dans les autres cas.} \end{cases} \tag{5.6}$$

Le paramètre θ est choisi le plus grand possible, mais tel que les valeurs $x'(k_{ij})$ et $x(k_{ij})$ avec $k_{ij} \in \gamma$ restent sur un même segment de linéarité de la fonction $\varphi_{ij}(x_{ij})$ qui fait partie de l'expression (4.4).

Ainsi,

$$\theta = \min_{k_{ij} \in \gamma} \delta(k_{ij}), \tag{5.7}$$

où

$$\delta(k_{ij}) = \begin{cases} d_{ij} - x(k_{ij}) \text{ avec } x(k_{ij}) > 0, \ k_{ij} \in l, \\ -x(k_{ij}) \text{ avec } x(k_{ij}) < 0, \ k_{ij} \in l, \\ d_{\lambda\mu} \text{ avec } k_{ij} = k_{\lambda\mu}. \end{cases} \tag{5.8}$$

La fonction $x'(k_{ij})$, $k_{ij} \in K'$, définie par les relations (5.6)-(5.8) est bien un programme du problème (4.1)-(4.4). En effet, la modification de la même valeur des composantes du programme X le long d'un certain itinéraire fermé orienté ne peut pas compromettre les égalités (4.1).

Les règles (5.7) et (5.8) ne permettent pas d'enfreindre les conditions (4.2). Pour ce qui est des restrictions (4.3), leur observation est assurée par la formule (5.6).

Le passage à un nouveau programme X' conduit à la modification de la fonction Φ du coût de transport total d'une valeur

$$\Phi(X') - \Phi(X) = \sum_{k_{ij} \in \gamma} [\varphi_{ij}(x'(k_{ij})) - \varphi_{ij}(x(k_{ij}))] =$$

$$= \sum_{\substack{x(k_{ij})>0 \\ k_{ij} \in l}} c_{ij}(x'(k_{ij}) - x(k_{ij})) -$$

$$- \sum_{\substack{x(k_{ij})<0 \\ k_{ij} \in l}} -c_{ji}(x'(k_{ij}) - x(k_{ij})) + c_{\lambda\mu}(x'(k_{\lambda\mu}) -$$

$$- x(k_{\lambda\mu})) = \theta \Big[\sum_{\substack{x(k_{ij})>0 \\ k_{ij} \in l}} c_{ij} - \sum_{\substack{x(k_{ij})<0 \\ k_{ij} \in l}} c_{ji} + c_{\lambda\mu} \Big].$$

Les égalités (5.3) avec $k_{ij} \in l$ entraînent ensuite

$$\Phi(X') - \Phi(X) = \theta \Big[\sum_{k_{ij} \in l} (u_j - u_i) + c_{\lambda\mu} \Big] =$$

$$= \theta (u_{\lambda_1} - u_\mu + u_{\lambda_2} - u_{\lambda_1} + \ldots + u_{\lambda_{s-1}} - u_{\lambda_{s-2}} + u_\lambda - u_{\lambda_{s-1}} + c_{\lambda\mu}) =$$

$$= \theta [c_{\lambda\mu} - (u_\mu - u_\lambda)].$$

Par hypothèse

$$c_{\lambda\mu} < u_\mu - u_\lambda.$$

Le paramètre $\theta > 0$ en tant que minimum de nombres positifs. Par conséquent,

$$\Phi(X') - \Phi(X) = \theta [c_{\lambda\mu} - (u_\mu - u_\lambda)] < 0. \tag{5.9}$$

On vérifie aisément que X' est un programme de base. Le système $K_{X'}$ des communications principales du programme X' peut contenir ou non la communication $k_{\lambda\mu}$. Dans le premier cas X' est naturellement un programme de base, car $K_{X'} \subset K_X$. Soit $k_{\lambda\mu} \in K_{X'}$.

S'il était possible de construire un itinéraire fermé γ' à partir des communications du réseau $(P, K_{X'})$, on aurait évidemment $k_{\lambda\mu} \in \gamma'$. Considérons l'itinéraire l' de p_μ à p_λ formé à partir de γ' par élimination de $k_{\lambda\mu}$. L'itinéraire l' est constitué de communications de l'ensemble K_X. Par suite, $l' = l$, où l est l'itinéraire de p_μ à p_λ construit au cours du passage au programme X', et toutes les communications de l appartiennent à $K_{X'}$. D'après la définition du nombre θ, une des communications de γ n'appartient pas d'avance à $K_{X'}$. Par démonstration ce ne peut être que $k_{\lambda\mu}$, ce qui contredit l'hypothèse. Il est donc impossible d'établir un itinéraire fermé à partir des communications de $K_{X'}$, ce qui signifie que X' est un programme de base.

L'hypothèse de la non-dégénérescence du problème $T(q, d, c)$ amène la conclusion que les ensembles K_X et $K_{X'}$ se composent d'un nombre égal de couples de communications opposées. C'est pourquoi le minimum (5.7) s'obtient sur une seule communication ; le couple correspondant de communications n'entre pas dans $K_{X'}$. Ainsi, ou bien $K_{X'}$ coïncide avec K_X, ou bien il se forme à partir de K_X en remplaçant l'un des couples de communications opposées par $k_{\lambda\mu}$ et $k_{\mu\lambda}$.

b) Lorsque c'est la deuxième condition du système (5.4)

$$u(p_\mu) - u(p_\lambda) < c_{\lambda\mu}, \quad x(k_{\lambda\mu}) = d_{\lambda\mu}$$

qui est compromise, on se trouve devant l'autre possibilité.

Dans ce cas, le programme X' se forme à partir de X en augmentant de $\theta > 0$ les transports le long des communications de l'itinéraire fermé γ_1 composé de $k_{\mu\lambda}$ et de l'itinéraire l_1 de p_λ à p_μ. L'itinéraire l_1 de p_λ à p_μ constitué de communications de K_X est défini d'après les mêmes règles que dans le cas a). L'itinéraire fermé γ_1 ne diffère de γ que par la direction.

Le nouveau programme X est calculé d'après les formules (5.6), (5.7) et (5.8), dans lesquelles l'itinéraire fermé γ est remplacé par γ_1. D'après ce qui a été démontré pour le cas a), X' est un programme de base du problème, et de plus

$$\Phi(X') - \Phi(X) = \theta\,[(u_\mu - u_\lambda) - c_{\lambda\mu}] < 0.$$

Les ensembles K_X et $K_{X'}$ des communications principales coïncident ou bien diffèrent par un couple de communications opposées.

Les deux autres cas de violation de (5.4) avec $i = \lambda$ et $j = \mu$ se ramènent aux cas a) et b) pour la communication $k_{\mu\lambda}$.

c) En effet, supposons que le premier membre de la première condition du système (5.4) ne soit pas observé :

$$-c_{\mu\lambda} > u(p_\mu) - u(p_\lambda), \quad x(k_{\lambda\mu}) = 0.$$

Des transformations évidentes permettent de mettre ces relations sous la forme

$$u(p_\lambda) - u(p_\mu) > c_{\mu\lambda}, \quad x(k_{\mu\lambda}) = 0,$$

ce qui signifie le cas a) pour la communication $k_{\mu\lambda}$.

d) Si, enfin, c'est la quatrième condition du système (5.4) qui est compromise, c'est-à-dire si

$$u(p_\mu) - u(p_\lambda) > -c_{\mu\lambda}, \; x(k_{\lambda\mu}) = -d_{\mu\lambda},$$

on fait appel à la communication $k_{\mu\lambda}$ pour aboutir au cas b) :

$$u(p_\lambda) - u(p_\mu) < c_{\mu\lambda}, \; x(k_{\mu\lambda}) = d_{\mu\lambda}.$$

Ainsi à la deuxième étape de l'itération on construit un itinéraire fermé γ qui passe par les points p_μ et p_λ et se compose de communications de l'ensemble $\{K_X, k_{\lambda\mu}, k_{\mu\lambda}\}$; de plus, dans les cas a) et d) l'itinéraire γ inclut $k_{\lambda\mu}$ alors que dans les cas b) et c), il contient $k_{\mu\lambda}$.

Le passage du programme de base X au programme de base X' se fait en augmentant les transports $x(k_{ij})$ avec $k_{ij} \in \gamma$, d'une valeur θ calculée d'après les formules (5.7) et (5.8). Le produit du paramètre θ par la valeur de la variable d'écart de la condition correspondante du système (5.4) avec $i = \lambda$, $j = \mu$ est égal à la différence $\Phi(X') - \Phi(X)$.

5.3. Toute itération qui contient l'étape 2 conduit à une décroissance de la fonction Φ. Par suite, au cours de la résolution du problème nous suivons différents programmes de base du problème considéré. Le nombre des programmes de base du problème $T(q, d, c)$ est fini. On peut s'en convaincre soit à l'aide de la proposition énoncée dans l'exercice 12, soit directement. Cette dernière voie est recommandée dans l'exercice 15.

Ainsi, lorsqu'on possède un programme de base initial du problème non dégénéré $T(q, d, c)$, la méthode des potentiels permet d'obtenir la solution en un nombre fini d'itérations.

Considérons l'un des moyens d'obtention d'un programme de base initial du problème. Ramenons le problème $T(q, d, c)$ à un problème équivalent ayant un point de fabrication p_0 et un point de consommation p_{N+1} en se guidant sur la règle exposée au § 4.1. Trouvons ensuite le flot maximal $\overline{X}$ dans le réseau $(\overline{P}, \overline{K})$ avec l'entrée p_0 et la sortie p_{N+1}. Deux cas sont alors possibles :

a) la valeur $\overline{x}_1$ du flot $\overline{X}$ est égale à

$$q = \sum_{q(p_i)>0} q(p_i);$$

b) la valeur $\overline{x}_1$ du flot $\overline{X}$ est inférieure à q.

Dans le premier cas les composantes du flot $\overline{X}$ associées aux communications $k_{ij} \in K'$ constituent un programme X du problème T (q, d, c). Le cas b) indique que le problème est non réalisable par suite de l'incompatibilité de ses conditions.

Il se peut que le programme X auquel on aboutit dans le cas a) ne soit pas un programme de base. S'il en est ainsi, le passage du programme X au programme de base X_1 s'effectue en décomposant successivement les itinéraires fermés dans le système des communications principales. Le processus de construction du programme de base X_1 repose sur les raisonnements donnés par la démonstration du théorème 4.3.

Une extension rigoureuse de la méthode des potentiels au cas des problèmes dégénérés impose l'introduction de la famille des ε-problèmes, comme c'est l'usage dans la discussion des formulations matricielles des problèmes de transport. On peut ainsi obtenir une règle de choix d'un couple de communications opposées à exclure du nombre des communications principales, règle éliminant toute possibilité d'un cyclage éventuel. Sans donner cette règle nous indiquerons une recommandation plus simple qui, sans garantir sûrement l'absence de cycle, le rend pratiquement impossible.

Dans le cas dégénéré, le rôle du système des communications principales du programme de base X passe au système des communications *de base* de ce programme qui compte $N - 1$ couples de communications opposées, contient toutes les communications principales du programme X et n'admet aucun itinéraire fermé. Dans le cas de non-dégénérescence, le système des communications principales coïncide avec le système des communications de base. Un programme de base dégénéré possède généralement plusieurs systèmes de communications de base, dont aucun ne coïncide avec l'ensemble des communications principales du programme. Nous conservons pour le système des communications de base du programme X la notation K_X.

L'ensemble $K_{X'}$ des communications de base du programme X' se forme à partir de l'ensemble $\{K_X, k_{\lambda\mu}, k_{\mu\lambda}\}$ par élimination de deux communications opposées $k_{\alpha\beta}$ et $k_{\beta\alpha}$ dont l'une permet de minimiser (5.7).

On choisit pour α et β n'importe quel couple d'indices i et j tels que

$$\delta(k_{ij}) = \theta.$$

Dans le cas dégénéré le programme de base peut évidemment rester le même pendant plusieurs itérations ($\theta = 0$) ; une seule itération ne fait que changer le système des communications de base de ce programme.

5.4. Voici un petit exemple numérique. Résoudre le problème de transport sur le réseau dont les conditions sont données par la figure 1.16.

Les communications du réseau ne sont pas marquées par les flèches, car les communications opposées k_{ij} et k_{ji} font ou ne font pas partie simultanément de l'ensemble K, c'est-à-dire $K = K'$. Chaque communication est matérialisée par un cercle dont la partie gauche indique sa capacité et la partie droite, le coût de transport; en outre, $d_{ij} = d_{ji}$, $c_{ij} = c_{ji}$. Le problème compte deux points de

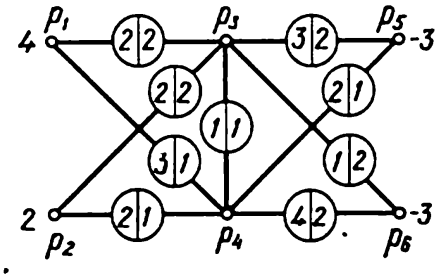

Fig. 1.16

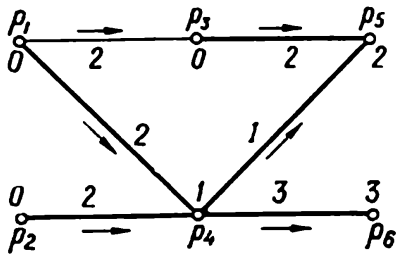

Fig. 1.17

fabrication (p_1 et p_2), deux points de consommation (p_5 et p_6) et deux points de transit (p_3 et p_4). Les valeurs non nulles de la fonction $q(p_i)$ sont placées à côté des points respectifs.

Adoptons comme programme initial le programme de base X représenté figure 1.17. Ce dessin ne représente que celles des communications du réseau dans lesquelles les transports prévus sont non nuls. Les valeurs des transports sont marquées à côté des communications correspondantes; leur direction est indiquée par des flèches. Les communications qui constituent le système K_X sont représentées par un gros trait.

Itération 1. Partons de l'étape 1 pour établir si X est un programme optimal ou non.

Constituons la suite des ensembles P_i en partant du point p_1:

$$P_1 = \{p_4\},\ P_2 = \{p_2,\ p_5,\ p_6\},\ P_3 = \{p_3\}.$$

Déterminons les valeurs des potentiels préalables:

$$u(p_1) = 0;\ u(p_4) = u(p_1) + c_{41} = 1;\ u(p_2) = u(p_4) - c_{42} = 0;$$
$$u(p_5) = u(p_4) + c_{45} = 2;\ u(p_6) = u(p_4) + c_{46} = 3;$$
$$u(p_3) = u(p_5) - c_{35} = 0.$$

Composons les différences des potentiels préalables pour les communications qui ne sont pas entrées dans le système K_X et vérifions si elles satisfont aux conditions (5.4). On a

$$\begin{aligned} u(p_3) - u(p_1) &= 0 < c_{13} = 2, & x(k_{13}) &= 2,\\ -1 = -c_{43} < u(p_4) - u(p_3) &= 1 = c_{34}; & x(k_{34}) &= 0,\\ -c_{32} < u(p_3) - u(p_2) &= 0 < c_{23}; & x(k_{23}) &= 0,\\ u(p_6) - u(p_3) &= 3 > c_{36} = 2; & x(k_{36}) &= 0. \end{aligned}$$

Les conditions (5.4) sont compromises pour les communications k_{13} et k_{33}. Il s'ensuit qu'il faut recourir à la deuxième étape. Choisissons la communication k_{13} [associée à une plus grande valeur de la variable d'écart pour la condition compromise (5.4)] et commençons l'étape 2. On est alors dans le cas b), où l'itinéraire fermé γ comportant les points p_1 et p_3 inclut la communication k_{31}.

Le système des ensembles P_i est déjà construit à partir du point p_1 et, de plus, dans le cas considéré, $s = 3$.

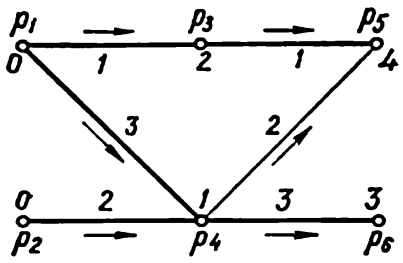

Fig. 1.18

L'itinéraire l de p_1 à p_3 comprend

$$k_{14},\ k_{45},\ k_{53}.$$

Selon les formules (5.7) et (5.8)

$$\delta_{14} = d_{14} - x(k_{14}) = 1,$$
$$\delta_{45} = d_{45} - x(k_{45}) = 1,$$
$$\delta_{53} = -x(k_{53}) = x(k_{35}) = 2,$$
$$\delta_{31} = x(k_{13}) = 2;$$
$$\theta = \min\{\delta_{14}, \delta_{45}, \delta_{53}, \delta_{31}\} = 1 = \delta_{14} = \delta_{45}.$$

Le nouveau programme X' s'obtient à partir de X en augmentant de 1 les transports le long des communications de l'itinéraire fermé γ. Les communications à éliminer de la base sont soit k_{14} et k_{41}, soit k_{45} et k_{54}. Laissons dans le système $K_{X'}$ les communications reliant les points p_1 et p_4. L'ensemble $K_{X'}$ se compose des communications marquées par un trait gros de la figure 1.18 qui correspond au programme X'.

Itération 2. Etape 1. Déterminons les valeurs des potentiels préalables:

$$u'(p_1) = 0, \quad u'(p_3) = 2, \quad u'(p_4) = 1,$$
$$u'(p_2) = 0, \quad u'(p_5) = 4, \quad u'(p_6) = 3.$$

Composons les différences des potentiels préalables pour les communications non incluses dans le système $K_{X'}$ et vérifions pour elles les conditions (5.4):

$$-2 = -c_{32} < u'(p_3) - u'(p_2) = 2 = c_{23}, \qquad x'(k_{23}) = 0,$$
$$-1 = -c_{43} = u'(p_4) - u'(p_3) = -1 < c_{34} = 1, \quad x'(k_{34}) = 0,$$
$$-2 = -c_{63} < u'(p_6) - u'(p_3) = 1 < c_{36} = 2, \quad x'(k_{36}) = 0,$$
$$u'(p_5) - u'(p_4) = 3 > c_{45} = 1, \quad x'(k_{45}) = d_{45} = 2.$$

Par conséquent, le programme X' est une solution du problème.

5.5. A titre de conclusion voici la description d'une autre méthode de résolution du problème $T(q, d, c)$, adhérente à la méthode des potentiels, mais qui, à la différence de cette dernière, diminue le coût de transport à chaque itération, indépendamment des pro-

priétés du problème exploré. La procédure de calcul dont le principe est exposé dans ce qui suit peut se nommer naturellement *méthode des itinéraires fermés.*

Soit $X = x(k_{ij})$, $k_{ij} \in K'$, un programme quelconque du problème $T(q, d, c)$ (pas nécessairement de base). Associons-lui l'ensemble R_X composé des communications $k_{ij} \in K'$ telles que

$$x(k_{ij}) < d_{ij}. \tag{5.10}$$

Déterminons sur les communications du réseau (P, R_X) la fonction C_X en posant

$$c_X(k_{ij}) = \begin{cases} c(k_{ij}) & \text{si } x(k_{ij}) \geqslant 0, \\ -c(k_{ji}) & \text{si } x(k_{ij}) < 0. \end{cases} \tag{5.11}$$

Nous conviendrons de dire que l'itinéraire fermé orienté γ du réseau (P, R_X) est négatif, nul ou positif, si la valeur $c_X(\gamma)$ de son coût de transport au sens de la fonction C_X est respectivement négative, nulle ou positive. La méthode consiste à améliorer successivement les programmes de transports.

Chaque itération de la méthode des itinéraires fermés consiste à construire d'après le programme X disponible un réseau de transport (P, R_X) et à définir sur les communications de ce réseau la fonction C_X.

La procédure permet d'obtenir un itinéraire fermé négatif du réseau (P, R_X) dont on se sert ensuite pour améliorer le programme X. L'absence d'itinéraires fermés négatifs constitués de communications du réseau (P, R_X) indique que le programme X est optimal.

Passons à un exposé plus détaillé de la méthode des itinéraires fermés. Celle-ci repose sur le critère d'optimalité suivant:

Si un réseau de transport (P, R_X) ne contient aucun itinéraire fermé négatif, le programme X est une solution du problème $T(q, d, c)$.

Enonçons la proposition auxiliaire suivante, nécessaire pour justifier ce critère d'optimalité.

L e m m e. *Une fonction non négative quelconque $z(k_{ij})$ définie sur les communications du réseau (S, R), non identiquement nulle et vérifiant les conditions*

$$z(p_i, S) = z(S, p_i), \quad p_i \in S,$$

peut être représentée sous la forme

$$z(k_{ij}) = \sum_{\lambda=1}^{t} z_\lambda(k_{ij}), \tag{5.12}$$

où

$$z_\lambda(k_{ij}) = \begin{cases} \theta_\lambda > 0 & \textit{avec } k_{ij} \in \gamma_\lambda, \\ 0 & \textit{dans les autres cas}; \end{cases}$$

γ_λ sont les itinéraires fermés orientés du réseau (S, R).

Nous laissons au lecteur la démonstration du lemme (cf. exercice 16).

En s'appuyant sur ce lemme, il est facile de s'assurer du bien-fondé du critère d'optimalité. En effet, supposons que le réseau (P, R_X) n'ait pas d'itinéraires négatifs, et soit $X' \neq X$ un programme quelconque du problème $T(q, d, c)$. Associons en un ensemble R les communications $k_{ij} \in K'$ telles que

$$x(k_{ij}) < x'(k_{ij}),$$

et posons

$$z(k_{ij}) = [x'(k_{ij}) - x(k_{ij})], \quad k_{ij} \in R; \quad S = P.$$

Il est clair que la fonction $z(k_{ij})$ vérifie les conditions du lemme; de plus, $R \subset R_X$.

En vertu de la convexité des fonctions $\varphi_{ij}(x_{ij})$ qui constituent la fonction du coût de transport Φ,

$$\Phi(X') - \Phi(X) \geqslant \sum_{k_{ij} \in R} c_X(k_{ij})\, z(k_{ij}). \tag{5.13}$$

En substituant ensuite à la fonction $z(k_{ij})$ du second membre de (5.13) sa représentation (5.12) on obtient

$$\Phi(X') - \Phi(X) \geqslant \sum_{\lambda=1}^{t} c_\lambda \theta_\lambda,$$

où c_λ est la valeur du coût de transport de l'itinéraire fermé γ_λ (dans le sens de la fonction C_X).

Par condition, $c_\lambda \geqslant 0$ pour tout λ (γ_λ se compose de communications de l'ensemble R_X). Par conséquent,

$$\Phi(X') - \Phi(X) \geqslant 0,$$

c'est-à-dire X est un programme optimal du problème $T(q, d, c)$.

Pour mettre en évidence les itinéraires fermés négatifs du réseau (P, R_X) il est rationnel d'utiliser un certain analogue de la méthode de Bellman-Shimbell (§§ 2.3-2.5).

En se donnant un $\varepsilon > 0$ suffisamment petit, introduisons dans la discussion la fonction $C_X(\varepsilon)$ dont les valeurs sont définies par la condition

$$c_X(k_{ij}, \varepsilon) = c_X(k_{ij}) + \varepsilon. \tag{5.14}$$

Introduisons ensuite la matrice $C_m(\varepsilon) = \| c_{ij}^{(m)}(\varepsilon) \|_N$ composée de valeurs du coût de transport des itinéraires généralisés m-optimaux du réseau (P, R_X), sur les communications duquel est donnée la fonction $C_X(\varepsilon)$ (pour abréger l'écriture on n'indique pas que $C_m(\varepsilon)$ est fonction de X).

Dans la mesure où il s'agit d'itinéraires *généralisés*, c'est-à-dire de suites de communications dont l'image géométrique est une ligne

brisée orientée qui, d'une manière générale, peut se couper elle-même, on peut affirmer (cf. § 2.5) que malgré la négativité de certaines valeurs de la fonction $C_X(\varepsilon)$ la formule

$$c_{ij}^{(m+n)}(\varepsilon) = \min_{1 \leqslant \mu \leqslant N} [c_{i\mu}^{(m)}(\varepsilon) + c_{\mu j}^{(n)}(\varepsilon)] \tag{5.15}$$

a lieu quels que soient m et n.

Il est aisé de remarquer que

$$c_{ij}^{(m)}(\varepsilon) = c_{ij}^{(m)} + \varepsilon \cdot \alpha_{ij}^{(m)},$$

où $c_{ij}^{(m)}$ est la valeur du coût de transport d'un itinéraire généralisé m-optimal de p_i à p_j dans le sens de la fonction C_X;

$\alpha_{ij}^{(m)}$, le plus petit nombre possible de communications dont puisse être constitué un tel itinéraire.

En utilisant la relation (5.15), construisons l'une après l'autre les matrices

$$C_1(\varepsilon),\ C_2(\varepsilon),\ C_4(\varepsilon),\ \ldots,\ C_{2^k}(\varepsilon),\ \ldots \tag{5.16}$$

Avant de s'attaquer à la composition de la matrice suivante $C_{2^{k+1}}(\varepsilon)$, il convient de vérifier deux conditions:

$$c_{ij}^{(2^k)}(\varepsilon) + c_{ji}^{(2^k)}(\varepsilon) \geqslant 0 \text{ pour tout } i > j, \tag{5.17}$$

$$k \leqslant \log_2 \left(\frac{N+1}{2}\right), \text{ où } N \text{ est le nombre de points du réseau.} \tag{5.18}$$

Le passage à la matrice $C_{2^{k+1}}(\varepsilon)$ ne s'effectue que lorsque ces deux conditions sont observées.

Nous dirons que le processus de prolongement de la suite (5.16) s'achève par le cas 1 si la condition (5.17) est compromise et par le cas 2 si (5.17) est observée tandis que (5.18) n'a pas lieu.

Examinons chaque cas séparément.

Le cas 1 entraîne l'existence d'un couple d'indices λ, μ tel que

$$c_{\lambda\mu}^{(2^k)}(\varepsilon) + c_{\mu\lambda}^{(2^k)}(\varepsilon) < 0. \tag{5.19}$$

L'inégalité (5.19) traduit la présence dans le réseau (P, R_X) d'un itinéraire fermé négatif γ; pour le mettre en évidence, il suffit d'avoir les matrices $C_1(\varepsilon)$ et $C_2^k(\varepsilon)$. Considérons le mode de construction de γ. Partant du point p_λ, composons la suite des points du réseau

$$p_\lambda,\ p_{\lambda_1},\ p_{\lambda_2},\ \ldots,\ p_{\lambda_r},\ \ldots, \tag{5.20}$$

en nous guidant sur la règle suivante: on pose que λ_{r+1} est la valeur de l'indice i qui assure

$$\min_{1 \leqslant i \leqslant N} [c_{\lambda_r i}^{(1)}(\varepsilon) + c_{i\mu}^{(2^k)}(\varepsilon)]. \tag{5.21}$$

(5.21) et l'égalité (5.15) entraînent avec $m = 1$ et $n = 2^k$ que quels que soient r et s ($s > r$),

$$c_X(k_{\lambda_r\lambda_{r+1}}) + c_X(k_{\lambda_{r+1}\lambda_{r+2}}) + \ldots +$$
$$+ c_X(k_{\lambda_{s-1}\lambda_s}) + \varepsilon(s-r) + c^{(2^k)}_{\lambda_s\mu}(\varepsilon) \leqslant c^{(2^k)}_{\lambda_r\mu}(\varepsilon). \qquad (5.22)$$

La suite (5.20) s'interrompt en un certain point p_{λ_s} si ce point y était déjà ($\lambda_s = \lambda_r$ avec $r < s$) ou si $p_{\lambda_s} = p_\mu$.

Dans le premier cas on prend comme itinéraire fermé γ la suite de communications

$$k_{\lambda_r\lambda_{r+1}},\ k_{\lambda_{r+1}\lambda_{r+2}},\ \ldots,\ k_{\lambda_{s-1}\lambda_r}.$$

La négativité de l'itinéraire γ se déduit de l'inégalité (5.22) avec $\lambda_r = \lambda_s$.

La deuxième possibilité conduit à l'itinéraire $l_{\lambda\mu}$ de p_λ à p_μ; de plus, d'après (5.22),

$$c_X(l_{\lambda\mu}) = c_X(k_{\lambda\lambda_1}) + \ldots + c_X(k_{\lambda_{s-1}\mu}) \leqslant c^{(2^k)}_{\lambda\mu}. \qquad (5.23)$$

Dans le deuxième cas il faut procéder à la construction de l'itinéraire $l_{\mu\lambda}$ de p_μ à p_λ en se guidant sur les règles déjà décrites. Il en résultera soit la mise en évidence d'un itinéraire fermé négatif γ, soit la constitution d'un itinéraire $l_{\mu\lambda}$ tel que

$$c_X(l_{\mu\lambda}) \leqslant c^{(2^k)}_{\mu\lambda}. \qquad (5.24)$$

Dans ce dernier cas, on prend pour γ l'itinéraire fermé constitué de $l_{\lambda\mu}$ et $l_{\mu\lambda}$, dont la négativité découle de (5.19), (5.23) et (5.24).

Le cas 2 indique l'absence d'itinéraires fermés négatifs dans le réseau (P, R_X). En effet, supposons qu'il n'en soit pas ainsi, c'est-à-dire qu'il existe un itinéraire fermé négatif γ_1 composé de communications de l'ensemble R_X. L'itinéraire γ_1 ne compte pas plus de N communications. Par conséquent, on peut le diviser en deux itinéraires l_{ij} et l_{ji}, dont chacun ne contient pas plus de $\frac{N+1}{2}$ communications. Etant donné que $2^k > \frac{N+1}{2}$, on a

$$C_X(l_{ij}) \geqslant c^{(2^k)}_{ij};\ c_X(l_{ji}) \geqslant c^{(2^k)}_{ji},$$

ou

$$c^{(2^k)}_{ij} + c^{(2^k)}_{ji} \leqslant c_X(l_{ij}) + c_X(l_{ji}) = c_X(\gamma_1) < 0.$$

L'inégalité obtenue contredit les conditions du cas 2 [les restrictions (5.17) sont compromises]. Ainsi, l'hypothèse de l'existence d'un itinéraire négatif γ_1 est incorrecte, ce qu'il fallait démontrer.

Supposons que la procédure de formation des matrices de la suite (5.16) se soit achevée par l'issue 1. Nous avons déjà vu qu'on peut alors dégager un itinéraire négatif γ constitué de communications de R_X.

Définissons le système de transport $X' = x'(k_{ij})$, $k_{ij} \in K'$, selon la formule

$$x'(k_{ij}) = \begin{cases} x(k_{ij}) + \theta \text{ avec } k_{ij} \in \gamma, \\ x(k_{ij}) - \theta \text{ avec } k_{ji} \in \gamma, \\ x(k_{ij}) \text{ dans les autres cas.} \end{cases} \tag{5.25}$$

Ici

$$\theta = \min_{k_{ij} \in \gamma} \delta(k_{ij}), \tag{5.26}$$

$$\delta(k_{ij}) = \begin{cases} d(k_{ij}) - x(k_{ij}) \text{ si } x(k_{ij}) \geqslant 0, \\ \quad\quad - x(k_{ij}) \text{ si } x(k_{ij}) < 0. \end{cases} \tag{5.27}$$

X' est évidemment un programme du problème $T(q, d, c)$, et de plus, d'après (5.26) et (5.27), les valeurs $x(k_{ij})$ et $x'(k_{ij})$ se situent sur le même segment de linéarité de la fonction $\varphi_{ij}(x)$.

Par conséquent,

$$\Phi(X') - \Phi(X) = c_X(\gamma) \cdot \theta. \tag{5.28}$$

Par condition, $c_X(\gamma) < 0$. L'itinéraire γ se compose de communications de l'ensemble R_X, c'est-à-dire

$$d(k_{ij}) - x(k_{ij}) > 0 \text{ pour } k_{ij} \in \gamma,$$

et le paramètre θ, qui est le minimum des nombres positifs $\delta(k_{ij})$, est positif.

En recourant à l'égalité (5.28), on obtient

$$\Phi(X') - \Phi(X) < 0,$$

ce qui signifie que le programme X' est plus économique que le programme X.

Supposons maintenant que la procédure de construction des matrices de la suite (5.16) ait conduit au cas 2. Puisque, comme nous l'avons montré, le réseau (P, R_X) est privé dans ce cas d'itinéraires fermés négatifs d'après le critère d'optimalité, le programme X est une solution du problème $T(q, d, c)$.

Ainsi, à chaque itération de la méthode des itinéraires fermés, on établit que le programme disponible est optimal, ou bien l'on construit un programme plus économique. Le programme initial du problème peut s'obtenir en appliquant la méthode du flot maximal ou toute autre méthode.

Puisque l'on suppose que les valeurs des fonctions Q, C et D sont entières, les grandeurs θ et $c_X(\gamma)$ sont des entiers, et (5.28) amène

$$\Phi(X') - \Phi(X) \leqslant -1.$$

L'inégalité obtenue indique que la méthode est finie. Cette conclusion reste valable quand les valeurs des fonctions qui définissent le problème sont rationnelles.

§ 6. Méthode hongroise

6.1. L'objet de ce paragraphe est la description d'une méthode de résolution du problème $T(q, d, c)$ qui est une extension de la méthode hongroise pour les problèmes T et T_d (§§ 5 et 6 du chapitre 14 de [87]). Cette méthode consiste en approximations successives de la solution cherchée à l'aide des programmes optimaux des problèmes $T(q_t, d, c)$ dont les fonctions q_t de fabrication et de consommation satisfont à la condition

$$\begin{aligned} &0 \leqslant q_t(p_i) \leqslant q(p_i) \quad \text{pour } p_i \in P_+, \\ &q(p_i) \leqslant q_t(p_i) \leqslant 0 \quad \text{pour } p_i \in P_-, \\ &q_t(p_i) = q(p_i) \qquad \text{pour } p_i \in P_0. \end{aligned} \tag{6.1}$$

La procédure de la résolution du problème fait décroître progressivement la variable d'écart

$$\varepsilon_t = \sum_{p_i \in P_+} [q(p_i) - q_t(p_i)] = \sum_{p_i \in P_-} [q_t(p_i) - q(p_i)]. \tag{6.2}$$

[L'égalité de la relation (6.2) se déduit de la dernière condition du système (6.1).]

Après quelques itérations on obtient $\varepsilon_t = 0$, ou bien l'on s'assure de l'impossibilité de poursuivre la diminution de la variable d'écart $\varepsilon_t > 0$. Dans le premier cas, le programme optimal du problème $T(q_t, d, c)$ est une solution du problème $T(q, d, c)$; dans le second, le problème étudié n'a pas de solution réalisable, les conditions de son système étant incompatibles. La méthode hongroise se fonde sur l'algorithme du flot maximal (§ 3) qui s'emploie pour réaliser la partie principale de chaque itération. L'exposé de la méthode suppose que le problème $T(q, d, c)$ est écrit sous la forme (4.1)-(4.4).

6.2. Voici la description d'une itération de la méthode hongroise. Les données disponibles avant le départ de l'itération sont les suivantes.

Le programme optimal $X_t = x_t(k_{ij})$, $k_{ij} \in K'$, du problème $T(q_t, d, c)$, où la fonction $q_t(p_i)$ satisfait à la condition (6.1). La fonction des potentiels $U_t = u_t(p_i)$, $p_i \in P$, du problème $T(q_t, d, c)$. Ces données sont soit obtenues à l'itération précédente, soit, dans le cas de la première itération, sont déterminées à l'étape préalable.

Le programme optimal X_t du problème $T(q_t, d, c)$ et la fonction de ses potentiels $u_t(p_i)$ sont liés par les conditions d'optimalité :

$$u_t(p_j) - u_t(p_i) = \begin{cases} c_{ij} & \text{si } 0 < x_t(k_{ij}) < d_{ij}, \\ -c_{ji} & \text{si } -d_{ji} < x_t(k_{ij}) < 0; \end{cases} \tag{6.3}$$

$$-c_{ji} \leqslant u_t(p_j) - u_t(p_i) \leqslant c_{ij} \text{ pour } x_t(k_{ij}) = 0; \tag{6.4}$$

$$\left.\begin{aligned} u_t(p_j) - u_t(p_i) &\geqslant c_{ij} \quad \text{pour } x_t(k_{ij}) = d_{ij}, \\ u_t(p_j) - u_t(p_i) &\leqslant -c_{ji} \text{ pour } x_t(k_{ij}) = -d_{ji}. \end{aligned}\right\} \tag{6.5}$$

k_{ij} est ici une communication quelconque de l'ensemble K'.

Toutefois, comme il a été déjà noté, le système des conditions (6.3)-(6.5) pour tout $k_{ij} \in K'$ est équivalent au système restreint qui consiste à choisir une seule communication de chacun des couples de communications opposées entrant dans K'.

Associons en un ensemble K_t les communications $k_{ij} \in K'$ telles que

$$u_t(p_j) - u_t(p_i) = c_{ij}. \tag{6.6}$$

D'habitude on entend par K'_t l'extension de l'ensemble K_t qui inclut un couple de communications opposées si l'une au moins des composantes de ce couple appartient à K_t. Soit $d_t(k_{ij})$, $k_{ij} \in K'_t$, le résultat de l'extension de la fonction $d(k_{ij})$, $k_{ij} \in K_t$, à l'ensemble K'_t, c'est-à-dire

$$d_t(k_{ij}) = \begin{cases} d(k_{ij}) & \text{si } k_{ij} \in K_t, \\ 0 & \text{si } k_{ij} \notin K_t,\ k_{ji} \in K_t. \end{cases} \tag{6.7}$$

Les données initiales de l'itération (programme X_t et fonction des potentiels U_t) engendrent un problème auxiliaire A_t, qui consiste à déterminer un système de transport $x(k_{ij})$, $k_{ij} \in K'$, qui vérifie les contraintes

$$\left.\begin{aligned} 0 \leqslant x(p_i, P) \leqslant q(p_i) &\text{ si } p_i \in P_+, \\ 0 \geqslant x(p_i, P) \geqslant q(p_i) &\text{ si } p_i \in P_-, \\ x(p_i, P) = q(p_i) = 0 &\text{ si } p_i \in P_0; \end{aligned}\right\} \tag{6.8}$$

$$-x(k_{ji}) = x(k_{ij}) \leqslant d_t(k_{ij}),\ k_{ij} \in K'_t, \tag{6.9}$$

$$-x(k_{ji}) = x(k_{ij}) = x_t(k_{ij}),\ k_{ij} \notin K'_t, \tag{6.10}$$

et maximise la fonction

$$x(P_+, P) = \sum_{p_i \in P_+} x(p_i, P). \tag{6.11}$$

Le problème auxiliaire A consiste ainsi à construire un système de transport dans les communications K'_t, qui

a) ne dépasse pas les valeurs de leurs capacités ;

b) assure avec le système de transports $x_t(k_{ij})$, $k_{ij} \notin K'_t$, le plus grand volume possible d'exportation à partir des points de fabrication compte tenu des possibilités de ces points;
c) ne dépasse pas la demande aux points de consommation et
d) équilibre le trafic aux points de transit.

Le problème A_t se ramène par le procédé usuel au problème $\bar{A}_t$ de la recherche d'un flot maximal dont certaines composantes sont fixées d'avance. A cette fin on adjoint à l'ensemble P l'entrée p_0 et la sortie p_{N+1}, ou on relie les points de fabrication p_j à p_0 par les communications k_{0j}, les points de consommation p_i à p_{N+1} par les communications $k_{i,\ N+1}$.

La fonction $d_t(k_{ij})$ est étendue aux communications nouvellement introduites à l'aide des égalités

$$d_t(k_{0j}) = q(p_j),\ p_j \in P_+;\ d_t(k_{i,\ N+1}) = -q(p_i),\ p_i \in P_-. \quad (6.12)$$

Le problème $\bar{A}_t$ consiste à rechercher un flot maximal dans le réseau $(\bar{P}, \bar{K}')$ à l'entrée p_0 et à la sortie p_{N+1}, compatible avec la fonction $d_t(k_{ij})$ pour $k_{ij} \in \bar{K}'_t$ et coïncidant avec le flot X_t des communications $k_{ij} \notin \bar{K}'_t$.

Ici

$$\bar{P} = \{P,\ p_0,\ p_{N+1}\},\ \bar{K} = \{K,\ k_{0j},\ k_{i,\ N+1}\},$$
$$\bar{K}_t = \{K\ ,\ k_{0j},\ k_{i,\ N+1}\}.$$

Comme flot initial du problème $\bar{A}_t$, on prend le système de transport $\bar{X}_t$, où

$$\bar{x}_t(k_{ij}) = \begin{cases} x_t(k_{ij}) & \text{pour } k_{ij} \in K', \\ x_t(p_j,\ P) & \text{pour } k_{ij} = k_{0j}, \\ -x_t(p_i,\ P) & \text{pour } k_{ij} = k_{i,\ N+1}. \end{cases} \quad (6.13)$$

Pour résoudre le problème $\bar{A}_t$, dont le flot initial $\bar{X}_t$ est connu, on peut appliquer l'algorithme du flot maximal exposé au § 3 en y introduisant au préalable l'évidente modification suivante. A chaque itération, le système des communications insaturées ne doit être formé que de communications de l'ensemble $\bar{K}_t$. De cette façon, le flot maximal dans le réseau $(\bar{P}, \bar{K}'_t)$ se construit à l'aide de l'algorithme du § 3. Etant donné que chaque communication k_{0j} a une capacité bornée, le problème $\bar{A}_t$ possède d'avance une solution et pour son exploration on ne peut appliquer que la partie principale de l'algorithme du § 3. Après quelques itérations (dont le nombre n'est généralement pas grand), on obtient un flot maximal $\bar{X}^*_t$ du problème $\bar{A}_t$. On obtient simultanément les ensembles $\bar{P}'_t$ et $\bar{P}''_t$ qui jouissent des propriétés suivantes:

$$p_0 \in \bar{P}'_t,\quad p_{N+1} \in \bar{P}''_t,\quad \bar{P}'_t \cap \bar{P}''_t = 0,\quad \bar{P}'_t \cap \bar{P}''_t = P; \quad (6.14)$$

$$\bar{x}^*_t(k_{ij}) = d_t(k_{ij}) \quad \text{si} \quad p_i \in \bar{P}'_t,\quad p_j \in \bar{P}''_t,\quad k_{ij} \in \bar{K}'_t. \quad (6.15)$$

Il est clair que le système de transport $X_t^* = \bar{x}_t^* (k_{ij})$, $k_{ij} \in K'$, est une solution du problème auxiliaire A_t. Posons

$$X_{t+1} = X_t^*; \quad q_{t+1}(p_i) = x^*(p_i, P), \quad p_i \in P.$$

D'après les conditions (6.8) du problème A_t, la fonction $q_{t+1}(p_i)$, $p_i \in P$, satisfait aux conditions (6.1); d'autre part,

$$\varepsilon_t - \varepsilon_{t+1} = \sum_{p_i \in P_+} x_{t+1}(p_i, P) - \sum_{p_i \in P_+} x_t(p_i, P) \geqslant 0,$$

puisque $X_{t+1} = X_t^*$ est une solution du problème A_t.

6.3. Si $\varepsilon_{t+1} = 0$, alors X_{t+1} est un programme du problème (4.1)-(4.4). On le tire des relations (6.1) et (6.2), dans lesquelles t est remplacé par $t + 1$. Le programme X_{t+1} et la fonction des potentiels U_t sont liés par les conditions (6.3)-(6.5). En effet, X_{t+1} ne diffère de X_t que dans les communications de l'ensemble K_t'. Il suffit donc de vérifier les conditions (6.3)-(6.5) pour $k_{ij} \in K_t$. La violation des conditions (6.3)-(6.5) pour un certain $k_{ij} \in K_t$ n'est possible que si

$$x_{t+1}(k_{ij}) < 0. \tag{6.16}$$

Mais alors $d_t(k_{ji}) > 0$, c'est-à-dire $k_{ji} \in K_t$, et par conséquent,

$$u_t(p_j) - u_t(p_i) = -c_{ji}. \tag{6.17}$$

Les relations (6.16)-(6.17) montrent que dans le cas considéré le passage de X_t à X_{t+1} n'enfreint pas les conditions (6.3)-(6.5).

Ainsi, le programme X_{t+1} satisfait aux conditions du critère d'optimalité et constitue une solution du problème $T(q, d, c)$. Nous appellerons cas 1 l'éventualité considérée ci-dessus. Supposons maintenant que $\varepsilon_{t+1} > 0$. Alors, il faut passer à une nouvelle fonction des potentiels $U_{t+1}^{(h)} = u_{t+1}^{(h)}(p_i)$, $p_i \in P$, en se guidant sur la formule

$$u_{t+1}^{(h)}(p_i) = \begin{cases} u_t(p_i) - h & \text{si } p_i \in P_t', \\ u_t(p_i) & \text{si } p_i \in P_t''. \end{cases} \tag{6.18}$$

P_t' et P_t'' désignent ici les ensembles obtenus à partir de $\bar{P}_t'$ et $\bar{P}_t''$ par élimination de l'entrée p_0 et de la sortie p_N respectivement [$\bar{P}_t'$ et $\bar{P}_t''$ s'obtiennent tout en recherchant la solution du problème $\bar{A}_t$ et vérifient les conditions (6.14)-(6.15)]. Le nombre $h > 0$ est choisi le plus grand possible, mais tel que X_{t+1} et $U_{t+1}^{(h)}$ soient associés par les conditions d'optimalité (6.3)-(6.5). Cherchons les contraintes justifiant le choix du nombre h. A cet effet écrivons les relations évidentes

$$u_{t+1}^{(h)}(p_j) - u_{t+1}^{(h)}(p_i) =$$

$$= \begin{cases} u_t(p_j) - u_t(p_i) \text{ soit si } p_i, p_j \in P_t', \text{ soit si } p_i, p_j \in P_t''; \\ u_t(p_j) - u_t(p_i) + h \text{ si } p_i \in P_t', p_j \in P_t''; \\ u_t(p_j) - u_t(p_i) - h \text{ si } p_i \in P_t'', p_j \in P_t'. \end{cases} \tag{6.19}$$

Supposons qu'au début $k_{ij} \in K_t$, $x_{t+1}(k_{ij}) \geqslant 0$. D'après la relation (6.15),

$$x_{t+1}(k_{ij}) = x^*_t(k_{ij}) = \begin{cases} d_t(k_{ij}) = d_{ij} \text{ si } p_i \in P'_t,\ p_j \in P''_t; \\ -d_t(k_{ji}) = 0 \text{ si } p_i \in P''_t,\ p_j \in P'_t. \end{cases} \quad (6.20)$$

Par conséquent, pour

$$0 < x_{t+1}(k_{ij}) < d_{ij},$$

les points p_i et p_j appartiennent ou n'appartiennent pas simultanément à l'ensemble P'_t, et (6.19) entraîne

$$u^{(h)}_{t+1}(p_j) - u^{(h)}_{t+1}(p_i) = u_t(p_j) - u_t(p_i) = c_{ij}.$$

Si $x_{t+1}(k_{ij}) = 0$, d'après (6.20), les inclusions $p_i \in P'_t$, $p_j \in P''_t$ entraînent l'égalité $d_{ij} = 0$ bien que par hypothèse $k_{ij} \in K_t \subset K$. Par suite *), ces inclusions ne peuvent avoir lieu, et d'après (6.19)

$$u^{(h)}_{t+1}(p_j) - u^{(h)}_{t+1}(p_i) \leqslant c_{ij}.$$

De façon analogue, pour $x_{t+1}(k_{ij}) = d_{ij}$,

$$u_{t+1}(p_j) - u_{t+1}(p_i) \geqslant c_{ij}.$$

Ainsi, pour un k_{ij} choisi, les conditions (6.3)-(6.5) restent vraies pour tout $h \geqslant 0$. Soit maintenant $k_{ij} \notin K'_t$, $x_{t+1}(k_{ij}) \geqslant 0$. Par hypothèse deux cas sont possibles

$$u_t(p_j) - u_t(p_i) < c_{ij}; \quad (6.21)$$

$$u_t(p_j) - u_t(p_i) > c_{ij}. \quad (6.22)$$

L'inégalité (6.21) a lieu avec $x_t(k_{ij}) = x_{t+1}(k_{ij}) = 0$; l'inégalité (6.22) avec $x_t(k_{ij}) = x_{t+1}(k_{ij}) = d_{ij}$. Dans le cas (6.21), la violation des conditions (6.3)-(6.5) pour $h > 0$ n'est possible qu'avec $p_i \in P'_t$, $p_j \in P''_t$. De façon analogue, dans le cas (6.22), le danger d'une violation des conditions d'optimalité pour $h > 0$ n'apparaît qu'avec $p_i \in P''_t$, $p_j \in P'_t$.

Associons en un ensemble S_t celles des communications $k_{ij} \in K$ qui n'entrent pas dans K'_t et qui vont du point $p_i \in P'_t$ au point $p_j \in P''_t$, de plus $x_{t+1}(k_{ij}) = 0$, ou bien qui joignent le point $p_i \in P''_t$ au point $p_j \in P'_t$, et dans ce cas $x_{t+1}(k_{ij}) = d_{ij}$. La procédure ultérieure de résolution du problème varie suivant que S_t comporte ou non au moins une communication.

Nous dirons que nous sommes dans le cas 2 si l'ensemble S_t est vide et dans le cas 3 pour une éventualité contraire.

Dans le cas 2, le programme X_{t+1} est associé à la fonction des potentiels par les conditions d'optimalité quel que soit $h \geqslant 0$. Montrons que les conditions du cas 2 rendent impossible la résolution du problème concerné.

---

*) L'ensemble K ne réunit que les communications k_{ij} pour lesquelles $d(k_{ij}) > 0$.

Introduisons le problème $\bar{A}$ du flot maximal dans le réseau $(\bar{P}, \bar{K})$ compatible avec la fonction $\bar{d}(k_{ij})$, $k_{ij} \in \bar{K}' = \{K, k_{0j}, k_{i,\,N+1}\}'$, où d'habitude,

$$\bar{d}(k_{ij}) = \begin{cases} d(k_{ij}) & \text{si } k_{ij} \in K', \\ q(p_j) & \text{si } k_{ij} = k_{0j},\ p_j \in P_+, \\ -q(p_i) & \text{si } k_{ij} = k_{i,\,N+1},\ p_i \in P_-; \end{cases}$$

$$\bar{P} = \{P, p_0, p_{N+1}\}.$$

Les problèmes $\bar{A}$ et $\bar{A}_t$ coïncident si $K_t = K$.

Considérons les ensembles $\bar{P}'_t = \{P'_t, p_0\}$ et $\bar{P}''_t = \{P''_t, p_{N+1}\}$ qui satisfont aux conditions (6.14), (6.15). Montrons qu'ils engendrent une coupe minimale du réseau $(\bar{P}, \bar{K})$ et que pour le problème $\bar{A}$ le flot $\bar{X}_{t+1} = \bar{X}^*_t$ est maximal. Etant donné que l'ensemble S_t est vide, on a pour $k_{ij} \notin K'_t$, $x_{t+1}(k_{ij}) \geqslant 0$,

$$\begin{aligned} \bar{x}_{t+1}(k_{ij}) &= d_{ij} \text{ si } p_i \in P'_t,\ p_j \in P''_t, \\ \bar{x}_{t+1}(k_{ij}) &= 0 \quad \text{si } p_i \in P''_t,\ p_j \in P'_t. \end{aligned} \tag{6.23}$$

Le système de conditions (6.14), (6.15) et (6.23) est une démonstration du fait constaté.

D'après l'énoncé du problème $\bar{A}$, le programme X_{t+1} du problème $T(q_{t+1}, d, c)$ est solution du problème qui consiste à maximiser la fonction (6.11) sous les conditions (6.8), et

$$-x(k_{ji}) = x(k_{ij}) \leqslant d_{ij},\ k_{ij} \in K'. \tag{6.24}$$

Par conséquent, tout système de transport X vérifiant les conditions (6.8) et (6.24) satisfait à l'inégalité

$$x(P_+, P) \leqslant x_{t+1}(P_+, P) = \sum_{p_i \in P_+} q(p_i) - \varepsilon_{t+1},$$

laquelle, compte tenu de la positivité de ε_{t+1}, peut se mettre sous la forme

$$x(P_+, P) < \sum_{p_i \in P} q(p_i). \tag{6.25}$$

Le problème $T(q, d, c)$ n'a pas de programmes, puisque supposer le contraire contredit l'inégalité (6.25).

Considérons le cas 3, où l'ensemble S_t compte au moins une communication. Posons

$$h_0 = \min_{k_{ij} \in S_t} |u_t(p_j) - u_t(p_i) - c_{ij}|, \tag{6.26}$$

$$u_{t+1}(p_i) = u_{t+1}^{(h_0)}(p_i),\ p_i \in P, \tag{6.27}$$

et passons à l'itération suivante en partant du programme X_{t+1} et de la fonction des potentiels U_{t+1}.

6.4. Reprenons brièvement la succession des opérations à l'intérieur d'une itération isolée de la méthode hongroise.

La fonction des potentiels U_t et le programme X_t du problème T (q_t, d, c) liés par les conditions (6.3)-(6.5) engendrent un problème auxiliaire A_t.

Le problème A_t se ramène au problème $\overline{A}_t$ du flot maximal avec des valeurs fixées de ses composantes x (k_{ij}) si $k_{ij} \notin K'_t$, et il se résout à l'aide de l'algorithme du § 3 en partant du flot initial $\overline{X}_t$ qui correspond au programme X_t. Les transports $\overline{x}^*_t$ (k_{ij}) (pour $k_{ij} \in K'$) du flot maximal $\overline{X}^*_t$ constituent une solution X^*_t du problème A_t, qui est considérée comme X_{t+1}. De plus,

$$\varepsilon_{t+1} \leqslant \varepsilon_t.$$

Si $\varepsilon_{t+1} = 0$ (cas 1), X_{t+1} est la solution cherchée du problème.

Si $\varepsilon_{t+1} > 0$, le processus de résolution se poursuit. On forme l'ensemble des communications S_t. S'il s'avère vide (cas 2), les conditions du problème T (q, d, c) sont contradictoires.

Si par contre S_t contient une ou plusieurs communications (cas 3), on construit d'après les formules (6.18), (6.26), (6.27) une nouvelle fonction des potentiels U_{t+1} et l'itération prend fin. Pour achever de justifier la méthode hongroise, il reste à montrer qu'elle est finie.

Montrons avant tout qu'après quelques itérations, ou bien la valeur de la variable d'écart $\varepsilon_{t+1} > 0$ diminue, ou bien l'on obtient le cas 2.

Si p_i et p_j appartiennent à l'ensemble P'_t avec $k_{ij} \in K'_t$, alors, d'après (6.19), la communication $k_{ij} \in K'_{t+1}$. Par conséquent, l'ensemble des points marqués $\overline{P}'_{t,1} = \{P'_{t,1}, p_1\}$ formé à la première étape de la résolution du problème $\overline{A}_{t+1}$ contient P'_t. En outre, cet ensemble inclut tous les points reliés à P'_t par les communications $k_{\lambda\mu}$ contenues dans S_t et minimisant l'expression (6.26). En effet, pour chacune de ces communications $k_{\lambda\mu}$,

$$u_{t+1}(p_\mu) - u_{t+1}(p_\lambda) = c_{\mu\lambda},$$

et de plus,

$$x_{t+1}(k_{\lambda\mu}) = \begin{cases} 0 & \text{si } p_\lambda \in P', \ p_\mu \in P'', \\ d_{\lambda\mu} & \text{si } p_\mu \in P', \ p_\lambda \in P'', \end{cases}$$

ce qui signifie que l'un des couples de communications opposées $k_{\lambda\mu}$, $k_{\mu\lambda}$ est une communication insaturée de l'ensemble $\overline{K}'_{t+1}$ dont l'origine est un certain point de l'ensemble P'_t.

Si l'itération considérée ne diminue pas la valeur de la variable d'écart, $X_{t+1} = X_{t+2}$, $P'_t \subset P'_{t,1} = P'_{t+1}$, et en outre P'_{t+1} est plus étendu que P'_t.

Ainsi, chacune des itérations qui ne diminue pas la valeur de la variable d'écart étend l'ensemble des points marqués. Par conséquent, si le cas 2 est impossible, la valeur de la variable d'écart diminue après N itérations au plus.

Supposons que les valeurs des fonctions $q(p_i)$, $p_i \in P$, et $d(k_{ij})$, $k_{ij} \in K$, soient des nombres entiers. Dans ce cas, chaque diminution de la variable d'écart la réduit au moins d'une unité. Pour finir la procédure il ne faut pas plus de ε_0 diminutions de la variable d'écart, où ε_0 est la variable d'écart du programme initial X_0. La diminution de la variable d'écart ou la mise en évidence de l'impossibilité de résoudre le problème imposant un nombre fini d'itérations font conclure que la méthode est finie.

6.5. Pour pouvoir s'attaquer à la résolution du problème $T(q, d, c)$ par la méthode hongroise, il faut construire au préalable un programme X_0 du problème $T(q_0, d, c)$ et une fonction des potentiels $u_0(p_i)$, $p_i \in P$ associés par les conditions d'optimalité (6.3)-(6.5), où la fonction $q_0(p_i)$, $p_i \in P$, satisfait aux restrictions (6.1). Le programme X_0 et la fonction des potentiels U_0 sont les données initiales pour la première itération; la procédure de leur formation constitue l'étape préparatoire de la méthode. Pour X_0 on adopte un programme nul. L'étape préparatoire consiste à former la fonction des potentiels U_0 associée à $X_0 = 0$ par les conditions d'optimalité (6.3)-(6.5). La fonction U_0 est recherchée par une méthode de récurrence.

Choisissons un point arbitraire du réseau (P, K), par exemple p_1, et posons

$$u_0(p_1) = 0.$$

Supposons que les valeurs de la fonction cherchée U_0 soient trouvées pour tous les points p_i des ensembles $P_0 \subset P_1 \subset \ldots \subset P_s$, où $P_0 = \{p_1\}$. Considérons un point quelconque $p_\mu \notin P_s$ relié au moins à un point de l'ensemble P_s à l'aide de communications de l'ensemble K' et posons $P_{s+1} = \{P_s, p_\mu\}$. Introduisons les ensembles de points $\varepsilon'_{\mu s}$, $\varepsilon''_{\mu s}$ en nous guidant par la règle:

$$p_\lambda \in \varepsilon'_{\mu s} \text{ si } p_\lambda \in P_s,\ k_{\lambda\mu} \in K',$$
$$p_\lambda \in \varepsilon''_{\mu s} \text{ si } p_\lambda \notin P_s,\ k_{\lambda\mu} \in K'.$$

Le choix de la valeur $u_0(p_\mu)$ de la fonction des potentiels U_0 est limité par les conditions

$$-c_{\lambda\mu} \leqslant u_0(p_\mu) - u_0(p_\lambda) \leqslant c_{\mu\lambda} \text{ si } p_\lambda \in \varepsilon'_{\mu s}; \tag{6.28}$$
$$-c_{\lambda\mu} \leqslant u_0(p_\mu) \leqslant c_{\mu\lambda} \text{ si } p_\lambda \in \varepsilon''_{\mu s}. \tag{6.29}$$

Le système d'inégalités (6.28)-(6.29) peut être résolu; l'une de ses solutions est $u_0(p_\mu) = 0$. En effet, avec $s = 0$ la proposition énoncée est évidente.

Supposons qu'elle soit vraie pour $P_0, P_1, \ldots, P_{s-1}$, les valeurs de la fonction $u_0(p)$ pour $p \in P_{s-1}$ ayant été déjà trouvées et vérifiant les conditions (6.28)-(6.29) où s est remplacé par $t = 0, 1, 2, \ldots, s-1$.

Pour prouver que (6.28)-(6.29) a une solution il suffit de montrer que

$$-c_{\lambda\mu} \leqslant -u_0(p_\lambda) \leqslant c_{\mu\lambda} \text{ si } p_\lambda \in \varepsilon'_{\mu s}. \tag{6.30}$$

Choisissons le numéro $s(\lambda) < s$ tel que

$$p_\lambda \notin P_{s(\lambda)},\ p_\lambda \in P_{s(\lambda)+1} = \{P_{s(\lambda)}, p(\lambda)\}.$$

(6.30) se déduit alors des conditions (6.29), où s est remplacé par $t = s(\lambda)$ et dont l'observation se déduit de l'hypothèse de récurrence. La compatibilité du système d'inégalités (6.28)-(6.29) est ainsi démontrée.

Posons

$$\left.\begin{aligned} u_0^+(p_\mu) &= \min\left\{\min_{\lambda\in\varepsilon'_{\mu s}} [c_{\mu\lambda} + u_0(p_\lambda)],\ \min_{\lambda\in\varepsilon''_{\mu s}} c_{\mu\lambda}\right\}, \\ u_0^-(p_\mu) &= \max\left\{\max_{\lambda\in\varepsilon'_{\mu s}} [-c_{\lambda\mu} + u_0(p_\lambda)],\ \max_{\lambda\in\varepsilon''_{\mu s}} (-c_{\lambda\mu})\right\}. \end{aligned}\right\} \tag{6.31}$$

Le système d'inégalités (6.28)-(6.29) est équivalent aux contraintes

$$u_0^-(p_\mu) \leqslant u_0(p_\mu) \leqslant u^+(p_\mu). \tag{6.32}$$

La valeur $u_0(p_\mu)$ est posée égale soit à $u_0^-(p_\mu)$, soit à $u_0^+(p_\mu)$, la préférence allant généralement à celle de ces deux valeurs qui répond au plus grand nombre de conditions (6.28)-(6.29) transformées en égalités par le $u_0(p_\mu)$ choisi. Après avoir défini $u_0(p_\mu)$, on poursuit le processus de construction de la fonction U_0 en partant de $P_{s+1} = \{P_s, p_\mu\}$. Comme le réseau (P, K) est supposé connexe, les potentiels $u_0(p_i)$ sont calculés pour tous les points $p_i \in P$ en $N-1$ étapes.

Les contraintes (6.28), respectées à chacune des étapes du stade préparatoire, conduisent au système d'inégalités

$$-c_{\mu\lambda} \leqslant u_0(p_\lambda) - u_0(p_\mu) \leqslant c_{\lambda\mu}, \text{ si } k_{\lambda\mu} \in K', \tag{6.33}$$

qui représente les conditions d'optimalité du programme $X_0 = 0$.

Le programme $X_0 = 0$ et la fonction des potentiels U_0 ainsi construite sont adoptés comme données initiales de la première itération. La variable d'écart ε_0 du programme X_0 est égale à la productivité totale de tous les points de fabrication du problème.

6.6. Pour une meilleure maîtrise de la méthode hongroise, résolvons à l'aide de cette méthode le problème exploré au § 5.4 (cf. fig. 1.16).

Stade préparatoire. Posons $u_0(p_1) = 0$. Le potentiel du point p_3 relié à p_1 par les communications du réseau satisfait aux contraintes :

$$|u_0(p_3)| \leqslant 2\ (k_{13}, k_{31}, k_{35}, k_{53}, k_{32}, k_{23}, k_{36}, k_{63}),$$

$$|u_0(p_3)| \leqslant 1\ (k_{34}, k_{43}),$$

où sont notées entre parenthèses les communications qui ont engendré l'inégalité correspondante. Dans le cas considéré, $u_0^+(p_3) = 1$, $u_0^-(p_3) = -1$. On pose $u_0(p_3) = 1$.

On détermine de façon analogue les autres valeurs de la fonction U_0

$$u_0(p_2) = -1,\ u_0(p_5) = 1,\ u_0(p_6) = 2,\ u_0(p_4) = 0.$$

Le réseau de transport (P, K_0), complété par l'entrée p_0, la sortie p_7 et les communications k_{01}, k_{02}, k_{57}, k_{67}, est représenté figure

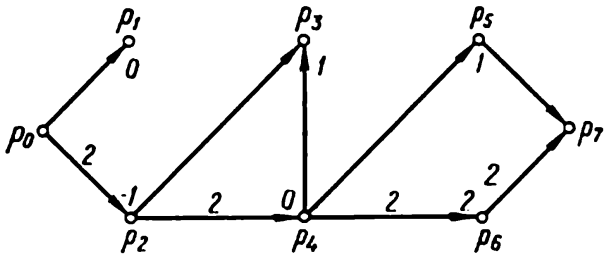

Fig. 1.19

1.19; près de chaque point du réseau on a porté la valeur correspondante de la fonction U_0.

Itération 1. Les données initiales de l'itération 1 sont le programme $X_0 = 0$ et la fonction des potentiels U_0. L'itération commence par la résolution du problème $\bar{A}_0$ de recherche du flot maximal dans le réseau $(\bar{P}, \bar{K}_0)$ représenté figure (1.19). Ce problème nécessite deux étapes de l'algorithme du flot maximal. A la première étape, deux unités de produit sont acheminées de p_2 à p_6 par p_4. La seconde étape établit la valeur maximale du flot obtenu; en outre $P_0' = \{p_1\}$. Le programme X_1 ne diffère du zéro que sur deux communications:

$$x_1(k_{24}) = x_1(k_{46}) = 2.$$

La variable d'écart ε_1 du programme obtenu X_1 se calcule suivant la formule

$$\varepsilon_1 = q(p_1) + q(p_2) - x_1(p_1, P) - x_1(p_2, P) = 4 > 0.$$

L'itération doit être poursuivie puisque le cas 1 n'a pas lieu. Le point p_1 est l'origine de deux communications k_{13} et k_{14} n'appartenant pas à K_0' qui se terminent en des points de l'ensemble P_0'' et sont prévues pour un transport nul. Entre les points P_0'' et p_1 il n'y a pas de transports égaux aux capacités des communications respectives. Donc, $S_0 = \{k_{13}, k_{14}\}$. Par conséquent nous sommes dans le cas 3, et de plus,

$$h = \min\{[c_{31} - (u_3 - u_1)],\ [c_{41} - (u_4 - u_1)]\} = \min\{1, 1\} = 1.$$

La nouvelle fonction des potentiels $u_1(p_i)$ ne diffère de $u_0(p_i)$ qu'en p_1: $u_1(p_1) = u_0(p_1) - h = -1$.

Itération 2. Le réseau de transport $(\overline{P}, \overline{K}_1)$ aux entrée p_0 et sortie p_7 est représenté figure 1.20. L'ensemble K_1 est obtenu à partir de K_0 par adjonction des communications k_{13} et k_{14}. L'algorithme du flot maximal du problème $\overline{A}_1$ se compose de trois étapes. A la première, deux unités de produit sont dirigées de p_1 à p_5 par p_4.

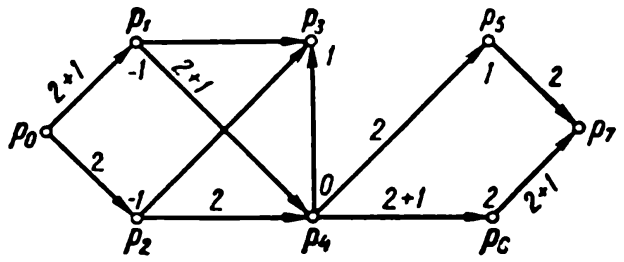

Fig. 1.20

A la deuxième, une unité de produit est acheminée de p_1 à p_6 par p_4. La troisième fixe le fait que le programme X_2 dont les transports sont indiqués figure 1.20 près de la notation des communications respectives, est solution du problème A_1. On construit en même temps l'ensemble $P'_1 = \{p_1, p_3\}$. Etant donné que

$$\varepsilon_2 = 1 > 0,$$

on poursuit l'itération. Construisons l'ensemble $S_1 = \{k_{35}, k_{36}\}$ et définissons

$$h = c_{53} - [u_1(p_5) - u_1(p_3)] = 2.$$

On a

$$u_2(p_i) = \begin{cases} u_1(p_i) \text{ si } p_i \notin P'_1, \\ u_1(p_i) - 2 \text{ si } p_i \in P'_1. \end{cases}$$

Itération 3. Le nouveau réseau de transport $(\overline{P}, \overline{K}_2)$ est représenté figure 1.21. L'ensemble K_2 se forme à partir de K_1 par

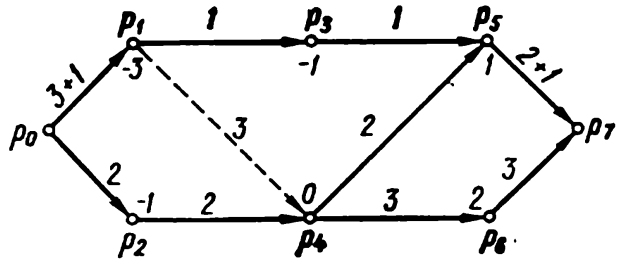

Fig. 1.21

élimination des communications k_{14}, k_{23}, k_{43} et adjonction de la communication k_{35}. La valeur de transport suivant la communication $k_{14} \notin K_2$, égale à 3, reste fixe au cours de toute l'itération (la communication k_{14} est tracée en pointillé). Le problème $\overline{A}_2$ (et, par consé-

quent, A_2) se résout en deux étapes de l'algorithme du flot maximal. A la première, une unité de produit est transportée de p_1 à p_5 par p_3. A la seconde, on constate l'optimalité du flot $\overline{X}_2^*$, dont les composantes sont indiquées à côté des communications correspondantes de l'ensemble K_2 et de la communication k_{14} sur la fig. 1.21. Comme la variable d'écart ε_3 du programme $X_3 = X_2^*$ est nulle, l'itération 3 s'achève par le cas 1 ; X_3 est la solution recherchée du problème.

On sait (cf. § 7.3 du chapitre 14 de [87]) que la méthode hongroise est à la base de plusieurs algorithmes de résolution des formulations matricielles des problèmes de transport. Les algorithmes de la méthode hongroise diffèrent seulement à l'étape de résolution du problème auxiliaire. On peut en dire de même de la méthode exposée dans le présent paragraphe. Les différents schémas de calcul de la méthode hongroise de résolution des problèmes de transport énoncés en termes de réseau sont liés aux modifications possibles de l'algorithme du flot maximal passées en revue au § 3.6. On peut obtenir notamment de là tous les algorithmes connus de résolution des problèmes T et T_d fondés sur la méthode hongroise.

Exercices

1. Montrer qu'un réseau de transport arbitraire se partage d'une façon et d'une seule en un nombre fini de sous-réseaux connexes maximaux.

2. Montrer que si dans l'algorithme de Minty (§ 2.2) on marque toutes les communications associées à la valeur minimale de la somme $h_\lambda + c_{\lambda\mu}$, tout itinéraire optimal issu du point p_1 est alors composé de communications marquées.

3. Etablir sur la base de l'algorithme de Minty une méthode de construction d'un itinéraire orienté entre les points fixés d'un réseau.

4. Etablir un algorithme pour définir tous les itinéraires orientés à extrémités fixées.

5. Montrer l'équivalence des égalités (2.7) et (2.10).

6. Prouver le bien-fondé du procédé de construction des itinéraires optimaux d'après la matrice C^*, exposé au § 2.4.

7. Montrer que si toutes les valeurs de la fonction D sont des nombres rationnels, l'algorithme du flot maximal est fini.

8. Enoncer le problème du flot maximal en termes de programmation linéaire, construire le problème dual et montrer qu'il est équivalent au problème de la recherche de la coupe minimale.

9. Montrer que la modification de l'algorithme du § 3, exposée au § 3.6, conduit à chaque itération à la construction d'un itinéraire de p_1 à p_N à degré d'insaturation maximal.

10. Soient A et B les ensembles des indices des points de fabrication et de consommation d'un problème quelconque du type T_d. Montrer que pour rendre possible la solution du problème T_d, il faut et il suffit que toute partition des ensembles A et B en A', A'' et B', B'' vérifie l'inégalité

$$\sum_{i \in A'} \sum_{j \in B'} d_{ij} \geqslant \sum_{i \in A''} a_i - \sum_{j \in B''} b_j.$$

Indication: se servir du théorème 3.1.

11. Appliquer les remarques qui précèdent l'énoncé du théorème 4.1, prouver le bien-fondé de cette proposition. Déduire le théorème 4.2 du théorème 4.1.

12. Montrer que la définition du programme de base donnée au § 4.4 est équivalente à la définition du programme de base du problème (4.5)-(4.7) considéré comme un problème de programmation linéaire à contraintes bilatérales.

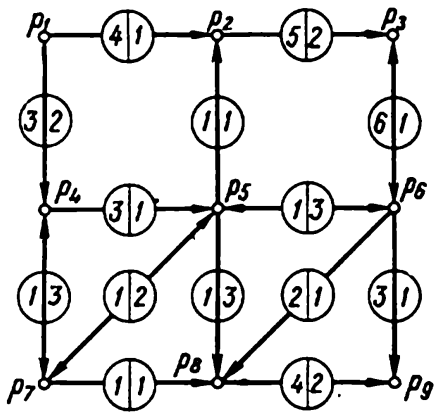

Fig. 1.22

13. Appliquer le théorème 4.3 pour obtenir une modification finie de l'algorithme du flot maximal, la fonction D étant quelconque.

14. Donner l'énoncé du problème du flot maximal dans un réseau à plusieurs entrées et sorties; composer son algorithme de résolution.

15. Montrer sans recourir aux résultats de la programmation linéaire que le nombre de programmes de base du problème $T(q, d, c)$ est fini.

16. Démontrer le lemme énoncé au § 5.5.

17. Trouver à l'aide de la méthode de Minty (§ 2.2) les itinéraires optimaux du point p_1 aux autres points du réseau représenté figure 1.22 (les valeurs du coût de transport $c(k_{ij})$ des communications du réseau sont portées à droite dans les cercles qui marquent les communications).

18. Appliquer la méthode de Bellman-Shimbell pour construire la matrice du coût de transport des itinéraires optimaux du réseau de l'exercice précédent.

19. Les valeurs des capacités $d(k_{ij})$ des communications du réseau représenté figure 1.22 sont placées à gauche dans les cercles correspondants. Construire le flot maximal du point p_1 au point p_9 de ce réseau.

20. Soit $T(q, d, c)$ le problème de transport sur le réseau de la figure 1.22 défini par les fonctions D et C dont les valeurs sont indiquées dans les cercles et par la fonction Q donnée par les conditions

$$q(p_i) = 0, \ i = 2, 4, 5, 6, 8;$$

$$q(p_1) = -q(p_3) = 2; \ q(p_7) = -q(p_9) = 3.$$

Résoudre le problème $T(q, d, c)$ en appliquant la méthode des potentiels

21. Résoudre le problème $T(q, d, c)$ par la méthode des itinéraires fermés (§ 5.5).

22. Résoudre le problème $T(q, d, c)$ par la méthode hongroise.

Chapitre 2

PROGRAMMATION LINÉAIRE ET THÉORIE DES JEUX

La programmation linéaire et la théorie des jeux sont nées des besoins pratiques en modèles et méthodes de planification économique et militaire. Les premières années, ces deux disciplines ont évolué indépendamment l'une de l'autre suivant des orientations propres à chacune d'elles. En 1951, G. B. Dantzig [34] établit une liaison remarquable entre la programmation linéaire et la théorie des jeux. L'entrelacement et l'interpénétration de ces disciplines mathématiques se sont avérés utiles pour chacune d'elles.

Ce qui nous intéresse dans le présent chapitre, ce n'est pas tant la théorie des jeux que la possibilité d'appliquer ses méthodes à la programmation linéaire. Pour cette raison, l'exposé des principes de la théorie des jeux au § 1 présente un caractère descriptif, et l'on n'attribue pas à la définition des notions fondamentales plus d'attention qu'il n'en est nécessaire pour saisir le sens de l'application des méthodes de solution des jeux au calcul des programmes optimaux des problèmes de programmation linéaire.

Le § 2 montre de quelle façon on peut faire correspondre à chaque jeu matriciel un problème de programmation linéaire et, inversement, comment on peut construire un jeu équivalent à un problème donné de programmation linéaire.

Le § 3 est consacré à l'exposé de certaines méthodes de résolution des jeux et des modifications possibles qui accélèrent leur convergence.

Le § 4 considère les méthodes itératives de résolution des problèmes de programmation linéaire. En outre, on y compare les méthodes finies et itératives et l'on donne les raisons qui font pencher en faveur de procédés combinatoires, permettant d'utiliser les mérites de l'une et de l'autre des méthodes de programmation linéaire.

§ 1. Généralités sur la théorie des jeux

1.1. L'objet de la théorie des jeux est l'étude des situations de conflit. Dans les situations de conflit, les intérêts des adversaires ne coïncident pas et parfois même sont opposés. De plus, chacun des

participants peut exercer une certaine influence sur la marche des événements, mais n'est pas capable de s'en rendre complètement maître.

Les modèles les plus simples de situations de conflit sont les jeux de société et de sport : cartes, échecs, dominos. La ressemblance extérieure des principes de comportement des participants aux jeux de société et aux opérations militaires, aux rencontres sportives et à la compétition économique justifie le nom de *jeux* donné aux modèles formalisés de situations de conflit et celui de *théorie des jeux*, aux méthodes mathématiques de leur discussion.

Les jeux sont conduits selon des règles déterminées. A l'issue d'un jeu, chaque participant obtient un certain gain (une perte, dans la terminologie de la théorie des jeux, est un gain exprimé en valeurs négatives). Les jeux diffèrent par le nombre de participants. Nous considérerons les jeux à deux personnes. Un jeu dans lequel les intérêts des joueurs sont directement opposés est appelé *jeu à somme nulle*. (Le gain d'un des joueurs est égal à la perte de l'autre.) Nous nous limiterons à l'exemple des jeux à somme nulle, bien qu'ils soient loin d'épuiser la classe des situations de conflit qui ont été étudiées.

Nous ne définirons pas ici d'une façon rigoureuse les notions de *partie*, *coup* et autres termes dont l'idée intuitive, liée aux jeux de société ou de sport, suffit pour la description générale du jeu. Les termes nécessaires pour l'exposé, mais qui ne sont pas essentiels à la compréhension des principes exposés dans ce chapitre, seront définis qualitativement sans entrer dans les détails. Le lecteur désireux de se familiariser encore plus avec les notions fondamentales de la théorie des jeux peut se reporter, par exemple, aux ouvrages très détaillés [48] et [50].

Certains jeux sont liés au hasard (donne des cartes, coup au but malgré les erreurs de tir, etc.). Pour formaliser les modèles de tels jeux, on introduit en plus des *coups personnels* des *coups aléatoires*. A chaque coup aléatoire, on attribue une distribution de probabilités des résultats possibles.

L'un des facteurs décisifs qui déterminent le comportement des joueurs est l'*information* dont disposent l'une et l'autre parties sur sa position et celle de l'adversaire, ainsi que sur le comportement passé des participants. Les échecs sont un *jeu avec information complète*. L'échiquier se trouve devant les adversaires, et un joueur attentif, avec une bonne mémoire, sait tout de la position dans laquelle il faut prendre une décision. Les dominos sont au contraire un *jeu avec information incomplète*. Le résultat d'un coup aléatoire — choix des pièces par le partenaire et par les adversaires — n'est pas connu du joueur.

Il est commode de représenter graphiquement les règles du jeu, la succession des coups personnels et aléatoires, l'information

à chaque coup et les résultats sous la forme de ce que l'on appelle une *arborescence*. Cette écriture détaillée du jeu s'appelle forme *développée* de jeu. La forme développée de jeu permet d'observer la dynamique du jeu et d'étudier sa structure et ses particularités. Cependant, l'abondance des classes de jeux développés rend malaisée l'édification d'une théorie unique d'exploration et de la résolution des jeux. Même les jeux les plus simples, sous forme développée imposent une écriture très compliquée et une discussion laborieuse.

L'introduction d'une des notions les plus importantes de la théorie des jeux — la notion de *stratégie* — permet de ramener les jeux développés les plus variés à une forme type unique, dite forme *normale* de jeu.

On appelle *stratégie* d'un jeu un système de règles qui déterminent univoquement le comportement d'un joueur à chaque coup en fonction de la situation définie par la marche du jeu. Le joueur ayant choisi une stratégie peut ne pas participer au jeu. Une personne neutre peut conduire le jeu suivant les instructions qu'il a données. Chacune des stratégies déterminées que peut choisir un joueur est appelée *stratégie pure* de ce joueur. Les stratégies pures n'épuisent pas toutes les possibilités des joueurs. Comme nous le verrons plus loin, il y a des situations dans lesquelles il est plus avantageux de choisir non une stratégie pure, mais la fréquence à laquelle il convient d'utiliser telle ou telle stratégie pure.

En se servant de la notion de stratégie, on peut considérer n'importe quel jeu de la façon suivante. Chaque joueur dispose d'un coup qui consiste à faire le choix d'une stratégie parmi celles qui appartiennent à un certain ensemble. Avec cela le joueur prend sa décision *sans disposer d'aucune information sur le choix de l'autre joueur*.

La réduction d'un jeu développé à la forme normale est généralement un problème de très grande taille.

Lorsque le jeu est à deux personnes, sa forme normale est dite *rectangulaire*. Un jeu rectangulaire à nombre fini de stratégies pures se nomme *matriciel*.

Dans ce qui suit nous ne considérerons que les *jeux matriciels à somme nulle*.

1.2. Supposons que le premier joueur dispose de m stratégies et le deuxième, de n stratégies. On admet de plus que si le premier joueur choisit la i-ème stratégie, et le deuxième la j-ième, le gain du premier (et donc la perte du second) est a_{ij}. La matrice $\| a_{ij} \|_{m,\, n}$ s'appelle *matrice de paiement* ou *matrice de gain*. Constatons que lors de la formalisation de situations de conflit réelles, la composition d'une matrice de paiement est une tâche complexe. Les principes qui définissent l'établissement d'une matrice de paiement se situent, d'une façon générale, en dehors de la théorie des jeux et relèvent de l'application concrète à laquelle est lié l'énoncé du problème.

La tâche de la théorie des jeux consiste à élaborer des principes qui déterminent le comportement des joueurs dans chaque situation de conflit concrète.

Il y a des jeux dans lesquels chacun des participants peut choisir une stratégie pure, comportement lui assurant un certain gain indépendamment de la conduite de l'adversaire et tel que les gains assurés des joueurs ne diffèrent que par le signe. Supposons que la stratégie correspondante du premier joueur ait le numéro i_0 et celle du deuxième le numéro j_0. Le gain du premier joueur est alors $a_{i_0j_0}$, et celui du deuxième $-a_{i_0j_0}$. Si le premier joueur choisit la stratégie i_0 alors que le deuxième refuse la stratégie j_0, le gain du premier ne peut qu'augmenter. En choisissant la stratégie j_0, le deuxième joueur empêche le premier de gagner plus que $a_{i_0j_0}$.

Ainsi,

$$a_{ij_0} \leqslant a_{i_0j_0} \leqslant a_{i_0j}, \quad i = 1, 2, \ldots, m; \; j = 1, 2, \ldots, n, \quad (1.1)$$

c'est-à-dire que le choix de la stratégie i_0 assure au premier joueur au moins un gain $a_{i_0j_0}$ et que l'utilisation de la stratégie j_0 par le deuxième joueur lui assure que le premier joueur ne recevra pas plus que $a_{i_0j_0}$.

On dit que dans de tels jeux les *stratégies pures* sont *optimales*. Un couple de nombres (i_0, j_0) qui correspond à des stratégies optimales des joueurs s'appelle *point-selle* de la matrice de paiement, et le nombre $a_{i_0j_0}$, *valeur du jeu*. [Le terme «point-selle » s'explique par l'interprétation géométrique des inégalités (1.1).] La matrice des gains peut avoir plusieurs points-selles qui tous déterminent toujours une valeur unique du jeu (cf. exercice 1).

Ainsi, *un jeu a une solution en stratégies pures* si l'on peut dégager dans la matrice de paiement un élément $a_{i_0j_0}$ qui vérifie les relations (1.1) pour tous i et j, c'est-à-dire si la matrice de paiement a un point-selle. Dans les jeux avec point-selle, chacun des joueurs peut s'assurer dans chaque partie, indépendamment du comportement de l'adversaire, une situation dite d'équilibre, c'est-à-dire une situation qu'il est naturel de considérer comme résultant d'un comportement intelligent des adversaires.

On a démontré que les échecs (ainsi que tous les jeux à information complète) se rapportent à la classe de jeux considérée, c'est-à-dire ont une solution en stratégies pures. Certes, la résolution de cette question de principe est encore fort loin de l'élaboration de recommandations concrètes garantissant l'obtention de la valeur du jeu. Toutefois il est très important, lors de l'étude du jeu, de savoir déterminer d'après son arborescence s'il admet des stratégies pures optimales. C'est qu'en effet les jeux sont loin de posséder tous cette propriété.

Un très grand nombre de situations de conflit rendent impossible l'existence de stratégies pures des joueurs susceptibles d'amener

un point d'équilibre indépendamment du comportement des adversaires.

La théorie permet cependant d'élaborer un comportement tel que le joueur qui l'observe dans chaque partie puisse s'assurer en moyenne (pour un grand nombre de parties) un point d'équilibre indépendamment du comportement de son adversaire. Si au cours de la réitération suivie des parties l'un des adversaires ne se tient pas aux règles de choix optimal des stratégies, le gain moyen de l'autre peut s'accroître. Mais chacun des joueurs peut toujours s'en tenir à des règles de choix des stratégies qui, dans le cas d'un grand nombre de parties, ne permettront pas à l'adversaire de dépasser un certain gain moyen.

Dans les jeux à information incomplète perd généralement son sens toute discussion sur une détermination du comportement le plus raisonnable des joueurs dans chaque partie. En règle générale, il est également impossible d'assurer un état d'équilibre indépendamment du comportement de l'adversaire si l'on applique dans les parties successives une suite déterminée de stratégies pures. Après un certain nombre de parties, l'adversaire apprendra les principes de notre comportement et en profitera pour choisir sa propre stratégie. La dissimulation du choix de notre comportement sera le mieux garantie dans le cas où pour choisir une stratégie dans chaque partie nous utiliserons un certain mécanisme aléatoire.

Ces considérations intuitives ramènent la question du choix du comportement optimal au choix des caractéristiques statistiques des mécanismes aléatoires qu'il faut adopter pour décider de la stratégie dans chaque partie. Autrement dit, le choix d'une stratégie optimale se ramène à celui de la fréquence à laquelle il faut recourir dans le jeu à chaque stratégie pure.

Le vecteur $U = (u_1, u_2, \ldots, u_m)$ dont chaque composante indique la fréquence (probabilité) relative à laquelle la stratégie pure correspondante est utilisée dans le jeu s'appelle *stratégie mixte* du premier joueur. La collection de nombres $W = (w_1, w_2, \ldots, w_n)$ est la stratégie mixte du second joueur. Il est évident que

$$u_i \geqslant 0,\ i = 1, 2, \ldots, m;\ w_j \geqslant 0,\ j = 1, 2, \ldots, n;$$
$$\sum_{i=1}^{m} u_i = 1;\quad \sum_{j=1}^{n} w_j = 1.$$

Une stratégie pure peut être définie comme une stratégie mixte dont toutes les composantes sont nulles sauf une.

Dans ce qui suit, les stratégies pures des deux adversaires s'écrivent sous la forme des vecteurs unitaires

$$e_i = (\overbrace{\underbrace{0, \ldots, 0, 1}_{i}, 0, \ldots, 0}^{m}) \text{ et } e_j = (\overbrace{\underbrace{0, \ldots, 0, 1}_{j}, 0, \ldots, 0}^{n})$$

respectivement.

La *stratégie optimale* d'un joueur est celle qui lui assure le plus grand gain moyen possible. On suppose que le jeu développé se fait sans ruse ni espionnage. Toute modification de l'information conduit à un nouveau jeu, pour lequel le comportement optimal sera autre.

Si le premier joueur choisit la i-ème stratégie pure avec une probabilité u_i, et le deuxième sa j-ième stratégie pure avec la probabilité w_j, le gain moyen du premier joueur est égal à

$$M(U, W) = \sum_{j=1}^{n} \sum_{i=1}^{m} a_{ij} u_i w_j = UAW,$$

où $A = \| a_{ij} \|_{m.\,n}$ est la matrice des gains (matrice de paiement du jeu).

Le gain du deuxième joueur est donc respectivement $-M$. La fonction $M(U, W)$ s'appelle *fonction de paiement*.

L'extension logique de la notion de solution d'un jeu au cas de stratégies mixtes conduit à la définition suivante. Nous dirons qu'un *jeu a une solution en stratégies mixtes* s'il existe des stratégies U^* et W^* et un nombre v tels que les stratégies mixtes U et W quelconques vérifient les relations

$$M(U, W^*) \leqslant v \leqslant M(U^*, W). \tag{1.2}$$

Posant $U = U^*$ et $W = W^*$, on obtient

$$v = M(U^*, W^*). \tag{1.3}$$

Le nombre v s'appelle *valeur du jeu*.

On voit sans peine que pour démontrer l'existence d'une solution du jeu, il suffit de justifier les inégalités (1.2) pour toutes les stratégies pures e_i et e_j.

En effet, si (1.2) se vérifie seulement pour les stratégies pures, on a pour U et W quelconques

$$M(U^*, W) = \sum_{j=1}^{n} w_j M(U^*, e_j) \geqslant v \sum_{j=1}^{n} w_j = v,$$

et de façon analogue,

$$M(U, W^*) \leqslant v.$$

La notion de point-selle concerne également le cas de stratégies mixtes. On dit qu'*un jeu a un point-selle* (U^*, W^*) si la fonction de paiement $M(U, W)$ vérifie les inégalités (1.2). Les propositions « le jeu a une solution » et « le jeu possède un point-selle » sont synonymes.

La théorie des jeux se sert souvent des notions de « maximin » et de « minimax ». Ces notions peuvent être appliquées aussi bien à l'ensemble de stratégies pures du jeu qu'à l'ensemble de stratégies mixtes. Introduisons-les d'abord pour l'ensemble de stratégies

pures. On a

$$\operatorname{minmax} \| a_{ij} \| = \min_j [\max_i a_{ij}],$$

$$\operatorname{maxmin} \| a_{ij} \| = \max_i [\min_j a_{ij}].$$

La définition de ces notions permet de montrer facilement que

$$\operatorname{maxmin} \| a_{ij} \| \leqslant \operatorname{minmax} \| a_{ij} \|.$$

Cette inégalité se transforme en égalité si et seulement si le jeu possède un point-selle en stratégies pures (exercice 2).

Dans un jeu où le choix de chacun des joueurs est limité à des stratégies pures, le gain assuré du premier joueur (qui ignore, naturellement, le choix de l'adversaire) est égal à maxmin $\| a_{ij} \|$. Le deuxième joueur, tout en ignorant le choix de l'adversaire, peut l'empêcher de gagner plus de minmax $\| a_{ij} \|$.

Ainsi, le gain du premier joueur, en fonction des stratégies pures adoptées par les deux joueurs, est compris, dans les conditions considérées, entre maxmin $\| a_{ij} \|$ et minmax $\| a_{ij} \|$.

Si par contre le premier joueur obtient une information sur les intentions de l'adversaire, son gain assuré devient minmax $\| a_{ij} \|$. Cela signifie que la différence

$$\sigma = \operatorname{minmax} \| a_{ij} \| - \operatorname{maxmin} \| a_{ij} \|$$

est l'accroissement du gain assuré du premier joueur dû à l'information sur le choix de l'adversaire. Pour le premier joueur, cette information ne présente un avantage que si le prix à payer pour elle n'excède pas la valeur σ. Dans un jeu à stratégies pures avec point-selle, une information sur le choix de l'adversaire ne change pas le gain assuré d'un joueur.

Supposons maintenant que le choix des joueurs ne soit pas limité à des stratégies pures. Définissons les notions de « maximin » et de « minimax » sur un ensemble de stratégies mixtes. Par définition

$$\operatorname{minmax} M = \min_{w_j} [\max_{u_i} \sum_{i=1}^{m} \sum_{j=1}^{n} a_{ij} u_i w_j],$$

$$\operatorname{maxmin} M = \max_{u_i} [\min_{w_j} \sum_{i=1}^{m} \sum_{j=1}^{n} a_{ij} u_i w_j]$$

avec

$$u_i \geqslant 0, \ i = 1, 2, \ldots, m; \quad w_j \geqslant 0, \ j = 1, 2, \ldots, n;$$

$$\sum_{i=1}^{m} u_i = 1; \quad \sum_{j=1}^{n} w_j = 1.$$

La définition de maxmin M et de minmax M permet facilement de démontrer l'inégalité

$$\text{maxmin } M \leqslant \text{minmax } M, \tag{1.4}$$

qui se transforme en égalité si et seulement si le jeu possède un point-selle (U^*, W^*) (exercice 3).

Maximin est la valeur assurée du gain moyen du premier joueur lorsqu'il ignore quelle stratégie mixte a choisie son adversaire. *Minimax* est le gain moyen assuré du premier joueur informé sur la stratégie mixte du deuxième joueur. L'inégalité (1.4) témoigne ainsi de ce qu'une information sur le choix de l'adversaire n'est pas inutile. Cependant, à la différence du cas où le choix se limitait à des stratégies pures, l'avantage assuré du joueur informé sur la stratégie mixte de l'adversaire se trouve être illusoire. Le théorème fondamental de la théorie des jeux affirme que tout jeu matriciel à deux personnes à somme nulle a toujours un point-selle en stratégies mixtes. Autrement dit, il existe toujours un nombre v et des vecteurs U^* et W^* tels que

$$M(U, W^*) \leqslant v = M(U^*, W^*) = \min_W \max_U M(U, W) =$$
$$= \max_U \min_W M(U, W) \leqslant M(U^*, W).$$

La procédure de calcul des stratégies optimales U^* et W^* et de la valeur du jeu v s'appelle *recherche de la solution du jeu* ou simplement *solution du jeu*.

§ 2. Relation entre jeux matriciels et programmation linéaire

2.1. Le présent paragraphe montre comment construire un couple de problèmes duals de programmation linéaire dont les solutions déterminent les stratégies optimales d'un jeu matriciel donné. Les paramètres des problèmes de programmation linéaire qui correspondent au jeu matriciel donné sont choisis ci-dessous au cours de la démonstration du théorème fondamental de la théorie des jeux. La construction d'un jeu matriciel d'après un problème donné de programmation linéaire est décrite au § 2.2. Les paramètres du jeu sont déterminés au cours de la démonstration du théorème 2.2.

Le § 2.3 fait l'inventaire des propriétés des stratégies optimales d'un jeu matriciel qui découlent des résultats théoriques fondamentaux de la programmation linéaire.

Théorème 2.1. (Théorème fondamental de la théorie des jeux). *Chaque jeu matriciel à somme nulle a une solution en stratégies mixtes, ce qui signifie qu'il existe des stratégies U^* et W^* respectivement du premier et du deuxième*

joueur telles que

$$M(U, W^*) \leqslant v = M(U^*, W^*) \leqslant M(U^*, W).$$

Démonstration. Soit la matrice du jeu égale à $\| a_{ij} \|_{m,\, n}$. On peut toujours considérer tous les éléments a_{ij} comme positifs. En effet, supposons que l'élément minimal de la matrice soit $\underline{a} < 0$. Augmentons tous les éléments de la matrice de paiement d'une quantité $a > -\underline{a}$. La fonction du gain correspondant à la matrice ainsi obtenue

$$\tilde{M}(U, W) = \sum_{i=1}^{m} \sum_{j=1}^{n} (a_{ij} + a)\, u_i w_j =$$

$$= M(U, W) + a \sum_{i=1}^{m} \sum_{j=1}^{n} u_i w_j = M(U, W) + a.$$

La fonction $\tilde{M}$ diffère de M d'une valeur constante. Aussi un point-selle de l'un des jeux est-il également point-selle de l'autre. On dit de tels jeux qu'ils sont à *équivalence stratégique*.

Considérons maintenant un couple de problèmes duals à matrice des conditions coïncidant avec la matrice de paiement $A = \| a_{ij} \|_{m,\, n}$ du jeu $(a_{ij} > 0)$, à vecteur des contraintes $B = (\underbrace{1, \ldots, 1}_{m})^T$ et à vecteur des coefficients de la forme linéaire $C = (\underbrace{1, \ldots, 1}_{n})$:

$$(C, X) \to \max, \tag{2.1}$$

$$AX \leqslant B, \tag{2.2}$$

$$X \geqslant 0 \tag{2.3}$$

et

$$(Y, B) \to \min, \tag{2.4}$$

$$YA \geqslant C, \tag{2.5}$$

$$Y \geqslant 0. \tag{2.6}$$

Tous les éléments de la matrice A sont positifs. L'ensemble convexe des conditions du problème (2.1)-(2.3) est donc borné. Le polyèdre (2.2)-(2.3) n'est pas vide (le vecteur $X = 0$ est un programme du problème). Par conséquent, le problème (2.1)-(2.3) a une solution réalisable. D'après le premier théorème de dualité (cf. [87], théorème 2.1, chapitre 6), le problème dual (2.4)-(2.6) a également une solution et

$$(C, X^*) = (Y^*, B),$$

où X^* et Y^* sont les solutions des problèmes respectifs.

Avec les vecteurs C et B que nous avons choisis, la dernière égalité signifie que

$$\sum_{j=1}^{n} x_j^* = \sum_{i=1}^{m} y_i^*.$$

La condition (2.5) entraîne

$$Y^* \neq 0.$$

Par conséquent,

$$(C, X^*) = (Y^*, B) = \sum_{j=1}^{n} x_j^* = \sum_{i=1}^{m} y_i^* = \frac{1}{v} > 0. \tag{2.7}$$

Posons

$$U^* = vY^*,$$
$$W^* = vX^*.$$

En vertu de (2.7), (2.3) et (2.6),

$$U^* \geqslant 0, \ W^* \geqslant 0;$$
$$\sum_{i=1}^{m} u_i^* = 1, \quad \sum_{j=1}^{n} w_j^* = 1.$$

D'autre part, puisque X^* et Y^* sont des programmes du couple de problèmes duals considéré, il vient

$$AW^* \leqslant vB, \tag{2.8}$$
$$U^*A \geqslant vC^T. \tag{2.9}$$

Multiplions à gauche les deux membres de l'inégalité (2.8) par un vecteur arbitraire $U \geqslant 0$ de dimension m tel que

$$(U, B) = \sum_{i=1}^{m} u_i = 1.$$

On obtient

$$UAW^* \leqslant v.$$

Multiplions à droite les deux membres de l'inégalité (2.9) par un vecteur arbitraire $W \geqslant 0$ tel que

$$(C, W) = \sum_{j=1}^{n} w_j = 1.$$

On obtient

$$U^*AW \geqslant v.$$

La comparaison des inégalités ainsi obtenues conduit à la relation

$$UAW^* \leqslant v \leqslant U^*AW,$$

ce qui signifie que U^* et W^* sont les stratégies optimales, et v la valeur du jeu à matrice de paiement A, ce qu'il fallait démontrer.

Au cours de la démonstration du théorème fondamental de la théorie des jeux, on fait correspondre au jeu matriciel à matrice de paiement $A = \| a_{ij} \|$ $(a_{ij} > 0)$ le couple suivant de problèmes duals de programmation linéaire:

$$\left.\begin{aligned} &\sum_{j=1}^{n} x_j \to \max, \\ &\sum_{j=1}^{n} a_{ij}x_j \leqslant 1, \quad i=1, 2, \ldots, m; \\ &x_j \geqslant 0, \quad j=1, 2, \ldots, n, \end{aligned}\right\} \tag{2.10}$$

et

$$\left.\begin{aligned} &\sum_{i=1}^{m} y_i \to \min, \\ &\sum_{i=1}^{m} a_{ij}y_i \geqslant 1, \quad j=1, 2, \ldots, n; \\ &y_i \geqslant 0, \quad i=1, 2, \ldots, m. \end{aligned}\right\} \tag{2.11}$$

Les composantes u_i^* et w_i^* des stratégies optimales du jeu sont associées aux composantes y_i^* et x_j^* des programmes optimaux des problèmes duals de programmation linéaire par les formules

$$w_j^* = \frac{x_j^*}{\sum\limits_{j=1}^{n} x_j^*}, \quad u_i^* = \frac{y_i^*}{\sum\limits_{i=1}^{m} y_i^*}.$$

La valeur du jeu

$$v = \frac{1}{\sum\limits_{j=1}^{n} x_j^*} = \frac{1}{\sum\limits_{i=1}^{m} y_i^*}.$$

Il est parfois plus commode de remplacer le couple de problèmes duals considéré par un autre couple de problèmes duals de programmation linéaire équivalent au jeu à matrice A (cf. exercice 18):

$$\begin{aligned} &v \to \min, \\ &(AW)_i \leqslant v, \ i = 1, 2, \ldots, m, \\ &\sum_{j=1}^{n} w_j = 1, \ w \geqslant 0, \end{aligned} \tag{2.12}$$

et

$$v \to \max,$$

$$(UA)_j \geqslant v, \; j = 1, 2, \ldots, n, \tag{2.13}$$

$$\sum_{i=1}^{m} u_i = 1, \quad U \geqslant 0.$$

Constatons que pour ramener un jeu matriciel à un couple de problèmes duals (2.10)-(2.11) il faut que les éléments a_{ij} de la matrice A du jeu soient positifs.

Par contre, la réduction du jeu au couple de problèmes duals (2.12), (2.13) n'impose aucune contrainte aux éléments de la matrice A. Toutefois, chacun des problèmes (2.12), (2.13) compte une variable et une contrainte de plus que les problèmes correspondants (2.10), (2.11).

2.2. Au cours de la démonstration du théorème fondamental de la théorie des jeux, on établit les paramètres du couple dual de problèmes de programmation linéaire dont la solution définit les stratégies optimales et la valeur optimale du jeu initial. On peut procéder également de façon inverse, en faisant correspondre à chaque couple de problèmes de programmation linéaire un jeu matriciel dont la valeur et les stratégies optimales permettent de calculer les programmes optimaux de problèmes duals (bien entendu, ces programmes existent). Il convient de souligner ici que tandis que les jeux matriciels ont toujours des stratégies optimales, les problèmes de programmation linéaire peuvent ne pas avoir de solution.

La correspondance entre les solutions d'un couple de problèmes duals et la solution d'un jeu est établie par le théorème 2.2 donné dans ce qui suit.

Soit A la matrice des conditions, B le vecteur des contraintes et C le vecteur des coefficients de la forme linéaire du problème de programmation linéaire.

Appelons les deux problèmes duals problèmes I et II et mettons-les sous la forme

$$\left.\begin{aligned} CX &\to \max, \\ AX &\leqslant B, \\ X &\geqslant 0 \end{aligned}\right\} \text{I}$$

et

$$\left.\begin{aligned} YB &\to \min, \\ YA &\geqslant C, \\ Y &\geqslant 0. \end{aligned}\right\} \text{II}.$$

Composons le jeu à matrice de paiement

$$\Pi = \begin{Vmatrix} 0 & A & -B \\ -A^T & 0 & C^T \\ B^T & -C & 0 \end{Vmatrix}. \tag{2.14}$$

La matrice Π est antisymétrique (le jeu correspondant est dit symétrique). Il est facile de s'assurer que la valeur v d'un jeu symétrique est nulle. En effet, soit U^* la stratégie optimale du premier joueur. En adoptant la même stratégie, le deuxième joueur rend nul le gain du premier joueur. Par conséquent, la valeur du jeu $v \leqslant 0$. Soit maintenant, la stratégie optimale W^* du deuxième joueur. En appliquant la même stratégie le premier joueur s'assure un gain nul. Par conséquent, $v \geqslant 0$. Ces deux appréciations de la valeur du jeu entraînent l'égalité $v = 0$.

Dans un jeu symétrique, les deux adversaires disposent des mêmes ensembles de stratégies. Il est facile de vérifier que les stratégies optimales pour le premier joueur le sont également pour son adversaire.

Un jeu symétrique justifie la proposition suivante.

Lemme 2.1. *Pour que la stratégie U^* soit stratégie optimale d'un jeu symétrique (c'est-à-dire pour qu'elle représente des stratégies optimales pour les deux joueurs), il faut et il suffit que toutes les stratégies W donnent lieu à l'inégalité*

$$M(U^*, W) \geqslant 0,$$

ou, ce qui revient au même, $U^*\Pi W \geqslant 0$, où Π est la matrice de paiement du jeu symétrique.

Démonstration. Soient U^* et W^* les stratégies optimales du premier et du deuxième joueur respectivement. Alors,

$$M(U^*, W) \geqslant M(U, W^*)$$

pour toutes les stratégies U et W. Notamment

$$M(U^*, W) \geqslant M(W^*, W^*) = 0.$$

Supposons inversement que $M(U^*, W) \geqslant 0$ pour toutes les stratégies W. Le jeu étant symétrique, $M(U, U^*) \leqslant 0$ pour toutes les stratégies U. D'où

$$M(U^*, W) \geqslant M(U^*, U^*) = 0 \geqslant M(U, U^*)$$

pour toutes les stratégies U et W. Cela signifie que la stratégie U^* est optimale pour chacun des joueurs.

Le lemme est ainsi démontré.

Corollaire. *Pour que la stratégie U^* soit stratégie optimale d'un jeu symétrique, il faut et il suffit que $U^*\Pi \geqslant 0$.*

Dans un jeu symétrique à matrice des gains (2.14), le nombre de stratégies pures de chaque joueur est $m + n + 1$.

Le théorème suivant établit une condition nécessaire et suffisante d'existence des solutions du couple de problèmes duals I et II énoncés en termes de jeu symétrique à matrice de paiement (2.14). Lorsque cette condition est remplie, on montre l'équivalence des problèmes I et II et du jeu à matrice (2.14).

Théorème 2.2. *Un couple de problèmes duals à matrice des conditions A et aux vecteurs B et C (définissant les deuxièmes membres des contraintes et les formes linéaires) a une solution si et seulement si le jeu à matrice de paiement* (2.14) *admet une stratégie optimale $U^* = (u_1^*, \ldots, u_{m+n}^*, u_{m+n+1}^*)$ telle que $u_{m+n+1}^* > 0$. Par ailleurs,*

$$y_i^* = \frac{u_i^*}{u_{m+n+1}^*}, \qquad i = 1, 2, \ldots, m;$$

$$x_j^* = \frac{u_{m+j}^*}{u_{m+n+1}^*}, \qquad j = 1, 2, \ldots, n.$$

Démonstration. 1. Soit les problèmes I et II aux programmes optimaux $X^* = (x_1^*, \ldots, x_n^*)^T$ et $Y^* = (y_1^*, \ldots, y_m^*)$ respectifs. Construisons le vecteur de dimension $(n + m + 1)$

$$U^* = \frac{1}{\alpha}(Y^*, X^{*T}, 1),$$

où

$$\alpha = 1 + \sum_{i=1}^{m} y_i^* + \sum_{j=1}^{n} x_j^* > 0.$$

On voit aisément que

$$U^*\Pi \geqslant 0.$$

En effet,

$$U^*\Pi = \frac{1}{\alpha}(Y^*, X^{*T}, 1) \begin{Vmatrix} 0 & A & -B \\ -A^T & 0 & C^T \\ B^T & -C & 0 \end{Vmatrix} =$$

$$= \frac{1}{\alpha}(-X^{*T}A^T + B^T,\ Y^*A - C,\ -Y^*B + X^{*T}C^T). \qquad (2.15)$$

En vertu des conditions du problème I

$$-X^{*T}A^T + B^T \geqslant 0.$$

En vertu des conditions du problème II

$$Y^*A - C \geqslant 0.$$

D'après le premier théorème de dualité,

$$-Y^*B + X^{*T}C^T = 0.$$

Par conséquent,

$$U^*\Pi \geqslant 0.$$

Ainsi les programmes optimaux du couple de problèmes duals déterminent une stratégie U^* qui assure l'observation de l'inégalité $U^*\Pi \geqslant 0$. D'après le corollaire du lemme 2.1, cela signifie que le vecteur U^* définit une stratégie optimale du jeu qui satisfait aux conditions du théorème ($u^*_{m+n+1} = \frac{1}{\alpha} > 0$).

2. Inversement, supposons maintenant que le vecteur $U^* = (u_1^*, \ldots, u^*_{m+n+1})$ soit stratégie optimale d'un jeu symétrique à matrice des gains Π et de plus, $u^*_{m+n+1} > 0$. Construisons les vecteurs X^* et Y^* de composantes

$$y_i^* = \frac{u_i^*}{u^*_{m+n+1}}, \qquad i = 1, 2, \ldots, m;$$

$$x_j^* = \frac{u^*_{m+j}}{u^*_{m+n+1}}, \qquad j = 1, 2, \ldots, n.$$

D'après le corollaire du lemme 2.1,

$$U^*\Pi \geqslant 0.$$

En utilisant ensuite la relation (2.15), on aboutit aux trois inégalités vectorielles:

$$-AX^* + B \geqslant 0, \tag{2.16}$$

$$Y^*A - C \geqslant 0 \tag{2.17}$$

$$-Y^*B + CX^* \geqslant 0. \tag{2.18}$$

(2.16) et (2.17) amènent que les vecteurs $X^* \geqslant 0$ et $Y^* \geqslant 0$ sont les programmes des problèmes duals I et II. En vertu du lemme 1.1 du chapitre 6 de [87], on a,

$$-Y^*B + CX^* \leqslant 0 \tag{2.19}$$

pour tout couple X^* et Y^* de programmes quelconques de problèmes duals de programmation linéaire.

Les inégalités (2.18) et (2.19) entraînent l'égalité

$$CX^* = Y^*B.$$

En raison du lemme 1.2 du chapitre 6 de [87], les vecteurs X^* et Y^* sont des programmes optimaux des problèmes I et II. Le théorème est donc démontré.

Ainsi, la résolution d'un couple de problèmes duals de programmation linéaire est ramenée au calcul des stratégies optimales d'un jeu symétrique équivalent, ayant une matrice de paiement d'ordre $m + n + 1$.

Dans le cas où la matrice des conditions A a tous ses éléments non négatifs et ne comporte pas de colonnes nulles, et où toutes les composantes des vecteurs B et C sont positives, on peut construire un jeu équivalent dont la matrice de paiement a n lignes et m colonnes.

(Remarquons que les problèmes de programmation linéaire pour lesquels sont remplies les conditions indiquées sont toujours résolubles.)

La matrice de paiement du jeu équivalent est

$$\widetilde{A} = \| \widetilde{a}_{ij} \|,$$

où

$$\widetilde{a}_{ij} = \frac{a_{ij}}{b_i c_j}.$$

Si $U^* = (u_1^*, \ldots, u_m^*)$ est la stratégie optimale du premier joueur, $W^* = (w_1^*, \ldots, w_n^*)$ la stratégie optimale du deuxième joueur et $v > 0$ la valeur du jeu, alors les collections de nombres

$$y_i^* = \frac{u_i^*}{b_i v} \quad \text{et} \quad x_j^* = \frac{w_j^*}{c_j v}$$

$$(i = 1, 2, \ldots, m; j = 1, 2, \ldots, n)$$

sont les solutions respectives du problème primal (I) et de son dual (II). Nous laissons au lecteur le soin de démontrer cette proposition (cf. exercices 4, 5).

Dans la deuxième représentation des problèmes de programmation linéaire sous forme de jeu équivalent les dimensions de la matrice de paiement sont plus petites que dans la première représentation. Dans une série de cas, il est pourtant plus commode d'utiliser la première représentation, car pour certaines méthodes de calcul il est plus avantageux de disposer d'un jeu sous forme symétrique.

2.3. A titre de conclusion à ce paragraphe, nous allons indiquer quelques propriétés des stratégies optimales d'un jeu matriciel. Partant des considérations théoriques de programmation linéaire exposées dans les chapitres 5 et 6 de [87], le lecteur pourra démontrer lui-même les propositions énoncées ci-dessous.

Voici d'abord quelques définitions. Une stratégie pure $e_k = (0, \ldots, 0, 1, 0, \ldots, 0)$ du premier joueur est dite *essentielle* s'il existe une stratégie mixte optimale $U^* = (u_1^*, \ldots, u_m^*)$ telle que sa k-ième composante u_k^* soit positive. Dans le cas contraire, la stratégie e_k est *inessentielle*. Les stratégies essentielles et inessentielles du deuxième joueur se définissent de façon analogue.

Une sous-matrice $\hat{A}$ de la matrice de paiement A est dite *essentielle* si elle se compose de ceux et seulement de ceux des éléments a_{ij} de la matrice A pour lesquels les stratégies pures e_i et e_j sont essentielles.

Voici les propriétés des stratégies optimales d'un jeu matriciel.

1°. Une stratégie pure e_k d'un joueur est essentielle si et seulement si elle permet d'obtenir la valeur du jeu quelle que soit la stratégie optimale de l'adversaire. En d'autres termes, la condition nécessaire et suffisante pour qu'une stratégie pure e_k du premier joueur soit essentielle est que l'égalité

$$e_k A W^* = v$$

soit vérifiée par toutes les stratégies optimales W^* du deuxième joueur. Si la stratégie e_k n'est pas essentielle, il existe toujours une stratégie optimale W^* du deuxième joueur qui justifie l'inégalité stricte

$$e_k A W^* < v.$$

La proposition ainsi énoncée est un analogue en théorie des jeux du deuxième théorème de dualité (théorème 2.2 du chapitre 6 de [87]), qui affirme que dans chaque couple de conditions duales de problèmes duals résolubles, l'une des conditions est lâche et l'autre serrée.

2°. Dans tout jeu matriciel, l'ensemble des stratégies optimales est un polyèdre convexe.

3°. Pour tout jeu matriciel, la différence entre les nombres $\overline{m}$ et $\overline{n}$ de stratégies pures essentielles du premier et du deuxième joueur est égale à la différence entre les dimensions ρ_u et ρ_w des polyèdres des stratégies optimales des deux adversaires:

$$\overline{m} - \overline{n} = \rho_u - \rho_w.$$

On en déduit que le nombre de stratégies essentielles des adversaires n'est le même qu'en cas d'égalité des dimensions des polyèdres respectifs des stratégies optimales.

4°. Pour chaque jeu matriciel il existe une sous-matrice carrée régulière $\| a_{i_\lambda j_\mu} \|$ de la matrice de paiement qui génère les systèmes d'équations

$$\sum_{\mu=1}^{t} a_{i_\lambda j_\mu} x_{j_\mu} = v, \qquad \lambda = 1, 2, \ldots, t;$$

$$\sum_{\mu=1}^{t} x_{j_\mu} = 1, \tag{2.20}$$

$$\sum_{\lambda=1}^{t} a_{i_\lambda j_\mu} y_{i_\lambda} = v, \qquad \mu = 1, 2, \ldots, t;$$

$$\sum_{\lambda=1}^{t} y_{i_\lambda} = 1. \tag{2.21}$$

Les solutions x_{j_μ}, y_{i_λ} des systèmes (2.20) et (2.21) déterminent les composantes des stratégies optimales w_j^* u_i^* des joueurs :

$$w_j^* = \begin{cases} x_{j_\mu} & \text{pour } j = j_\mu, \\ 0 & \text{pour } j \neq j_\mu, \end{cases} \quad \mu = 1, 2, \ldots, t;$$

$$u_i^* = \begin{cases} y_{i_\lambda} & \text{pour } i = i_\lambda, \\ 0 & \text{pour } i \neq i_\lambda, \end{cases} \quad \lambda = 1, 2, \ldots, t.$$

La proposition 4° est un corollaire du théorème sur l'existence d'une solution de base du problème de programmation linéaire (cf. le théorème 4.2 du chapitre 5 de [87]), appliqué à la représentation d'un jeu sous la forme d'un couple de problèmes duals (2.12), (2.13).

En modifiant légèrement l'énoncé de cette dernière proposition, on obtient le théorème de Shapley-Snow définissant les stratégies optimales d'un jeu [82].

Le théorème de Shapley-Snow établit l'ordre suivant de calcul des stratégies optimales.

1. Choisir une sous-matrice carrée régulière quelconque $\| a_{i_\lambda j_\mu} \|_t$ de la matrice A du jeu et calculer à partir des systèmes (2.20), (2.21) les valeurs de x_{j_μ}, y_{i_λ} et v.

2. Vérifier si les conditions

$$x_{j_\mu} \geqslant 0, \; y_{i_\lambda} \geqslant 0, \; \mu, \lambda = 1, 2, \ldots, t;$$

$$\sum_{\mu=1}^{t} a_{ij_\mu} x_{j_\mu} \leqslant v, \qquad i = 1, 2, \ldots, m;$$

$$\sum_{\lambda=1}^{t} a_{i_\lambda j} y_{i_\lambda} \geqslant v, \qquad j = 1, 2, \ldots, n$$

sont respectées.

Si toutes les conditions considérées sont satisfaites, la collection des composantes x_{j_μ}, y_{i_λ} définit une stratégie optimale et le nombre v, la valeur du jeu. Si non, il faut passer à une autre sous-matrice. Le théorème de Shapley-Snow ramène ainsi la résolution du jeu à matrice de paiement A à la révision non ordonnée des systèmes de $m + 1$ vecteurs des conditions du problème (2.12) ou de $n + 1$ vecteurs des conditions du problème (2.13) (c'est-à-dire des sous-matrices carrées de la matrice du jeu). Pour un jeu au nombre tant soit peu important de stratégies pures, la méthode de résolution résultant du théorème de Shapley-Snow n'est pas efficace.

§ 3. Méthodes de résolution des jeux

3.1. La méthode de calcul des stratégies optimales fondée sur le théorème de Shapley-Snow est une méthode finie qui assure l'obtention d'une solution exacte du jeu en un nombre fini d'opérations. Pourtant comme nous l'avons vu, elle n'est valable pratiquement que pour les jeux à très petit nombre de stratégies. Il semble que les méthodes les plus efficaces de calcul des valeurs exactes des composantes des stratégies optimales soient les méthodes finies de programmation linéaire. Pour résoudre le jeu à matrice de paiement positive $A = \| a_{ij} \|_{m,\, n}$ (il est toujours possible de se ramener à ce cas), il suffit, comme on l'a montré au paragraphe précédent, de résoudre le problème de programmation linéaire qui consiste à minimiser la forme linéaire

$$\sum_{i=1}^{m} y_i$$

sous les conditions

$$\sum_{i=1}^{m} a_{ij} y_i \geqslant 1, \quad j = 1, 2, \ldots, n;$$

$$y_i \geqslant 0, \; i = 1, 2, \ldots, m.$$

Les composantes u_i^* de la stratégie optimale du premier joueur sont déterminées à l'aide des composantes y_i^* de la solution du problème de programmation linéaire d'après la formule

$$u_i^* = \frac{y_i^*}{\sum\limits_{i=1}^{m} y_i^*}, \quad i = 1, 2, \ldots, m.$$

Les composantes w_j^* de la stratégie optimale du deuxième joueur se calculent à partir des composantes x_j^* de la solution du problème dual selon les formules analogues

$$w_j^* = \frac{x_j^*}{\sum\limits_{j=1}^{n} x_j^*}, \quad j = 1, 2, \ldots, n.$$

Ainsi, les méthodes de programmation linéaire peuvent être utilisées pour la résolution des jeux sans mise au point particulière.

A la différence de la méthode de Shapley-Snow, qui conduit à une révision non ordonnée des sous-matrices carrées régulières de la matrice du jeu, les méthodes finies de programmation linéaire interviennent dans la solution du jeu par une révision ordonnée et économique de ces sous-matrices.

Remarquons que pour résoudre un jeu à l'aide des méthodes finies de programmation linéaire, il n'est pas toujours avantageux de rame-

ner la matrice de paiement à une matrice positive équivalente. Tous les raisonnements effectués plus haut pour les jeux à matrice de paiement positive peuvent être repris, par exemple, pour les jeux à matrice non négative dont chaque colonne compte au moins un élément positif (on assure en outre que $v > 0$). Même si parmi les éléments des matrices il y a des nombres négatifs, les matrices de paiement à grand nombre de zéros ne doivent pas être ramenées à des matrices positives. La résolution des problèmes correspondants de programmation linéaire ne peut que devenir plus complexe. Il est rationnel, dans de tels cas, de ramener le jeu au couple de problèmes duals (2.12), (2.13).

Tableau 2.1

	N	C_x	B_x	c_0	c_1	c_2	c_3	A_k	Θ	N itérations
	1		A_4	1	1			1	1	
←	2		A_5	1		1		$\boxed{4}$	1/4	0
	3		A_6	1			1	2	1/2	
	4	—	—					—1	—	
	1		A_4	3/4	1	—1/4		2	3/8	
→	2	1	A_1	1/4		1/4			—	1
←	3		A_6	1/2		—1/2	1	$\boxed{3}$	1/6	
	4	—	—	1/4		1/4		—1	—	
←	1		A_4	5/12	1	1/12	—2/3	$\boxed{37/12}$	5/37	
	2	1	A_1	1/4		1/4		1/4	1	2
→	3	1	A_2	1/6		—1/6	1/3	—1/6	—	
	4	—	—	5/12		1/12	1/3	—11/12	—	
→	1	1	A_3	5/37	12/37	1/37	—8/37			3
	2	1	A_1	8/37	—3/37	9/37	2/37			
	3	1	A_2	7/37	2/37	—6/37	11/37			
	4	—	—	20/37	11/37	4/37	5/37			

Tableau 2.2

N	V	A_1	A_2	A_3	A_4	A_5	A_6
1	1	1	2	3	1		
1	1	4		1		1	
3	1	2	3				1
4	c_j	1	1	1			
0	Δ	$\boxed{-1}$	-1	-1			
1	Δ'		$\boxed{-1}$	$-3/4$		$1/4$	
2	Δ''			$\boxed{-11/12}$	$11/37$	$1/12$	$1/3$
3	Δ'''					$4/37$	$5/37$

A titre d'exemple, résolvons à l'aide du deuxième algorithme de la méthode du simplexe le jeu à matrice de paiement

$$\begin{Vmatrix} 1 & 2 & 3 \\ 4 & 0 & 1 \\ 2 & 3 & 0 \end{Vmatrix}.$$

Le problème correspondant de programmation linéaire se présente sous la forme

$$\begin{aligned} L(X) = x_1 + x_2 + x_3 \quad &\to \max, \\ x_1 + 2x_2 + 3x_3 &\leqslant 1, \\ 4x_1 + \quad + \ x_3 &\leqslant 1, \\ 2x_1 + 3x_2 \quad &\leqslant 1, \\ x_1 \geqslant 0,\ x_2 \geqslant 0,\ x_3 &\geqslant 0. \end{aligned}$$

L'ordre de résolution du problème est donné dans les tableaux principal 2.1 et auxiliaire 2.2. La solution s'obtient en trois itérations. Le programme optimal du primal s'écrit

$$X^* = \left(\frac{8}{37}, \frac{7}{37}, \frac{5}{37}\right).$$

La solution de son dual (estimation des conditions du problème initial) est donnée par le vecteur

$$Y^* = \left(\frac{11}{37}, \frac{4}{37}, \frac{5}{37}\right).$$

La valeur optimale de la forme linéaire

$$L(X^*) = \frac{20}{37}.$$

On en tire les stratégies optimales

$$U^* \simeq (0{,}55;\ 0{,}20;\ 0{,}25); \quad W^* \simeq (0{,}40;\ 0{,}35;\ 0{,}25).$$

La valeur du jeu est

$$v = \frac{1}{L(X^*)} \simeq 1{,}85.$$

3.2. Nous allons examiner ci-dessous les méthodes de résolution des jeux nées de tentatives directes de calcul des stratégies optimales. Ce sont des méthodes itératives qui n'assurent qu'une résolution approchée des jeux (avec une approximation quelconque donnée d'avance). Les erreurs dont s'accompagne le calcul des éléments d'une matrice de paiement dans les problèmes pratiques justifient l'application de méthodes approchées.

Considérons la méthode itérative de résolution des jeux proposée par G. W. Brown [8] et justifiée par J. Robinson [66]. Cette méthode, très simple mais insuffisamment efficace, porte le nom de méthode du « jeu fictif ».

Supposons qu'un jeu à matrice $A = \| a_{ij} \|_{m \cdot n}$ se répète de multiples fois et que les joueurs, en retenant les stratégies adoptées par chaque adversaire dans chaque partie, évaluent les gains ainsi obtenus et perfectionnent peu à peu leur maîtrise. Ce perfectionnement peut s'effectuer de la manière suivante d'une partie à l'autre.

Dans la première partie, le premier joueur, n'ayant pas d'expérience ni de connaissances, choisit une stratégie pure quelconque. Soit i_1 son numéro. Nous considérerons cette stratégie pure comme une stratégie mixte $U^{(1)}$ à composante non nulle unique. Le deuxième joueur, sachant le choix de l'adversaire, choisit la meilleure stratégie possible par rapport à $U^{(1)}$. Autrement dit, le deuxième joueur adopte la stratégie pure de numéro j_1 associée au plus petit élément $a_{i_1 j_1}$ de la ligne i_1 de la matrice de paiement A. Appelons cette stratégie $W^{(1)}$. Sur ce, le premier joueur, ayant oublié les raisons qui incitèrent son adversaire à choisir la stratégie $W^{(1)}$, choisit la stratégie pure i_2, la meilleure par rapport à j_1 de l'adversaire. Le deuxième joueur, en prenant sa décision, sait que son adversaire a choisi dans une partie la stratégie pure i_1, et dans l'autre la stratégie pure i_2. Il suppose que son adversaire s'en tient à la stratégie mixte $U^{(2)}$ composée de stratégies pures i_1 et i_2, chacune avec la probabilité 1/2. Sous cette hypothèse, le deuxième joueur doit adopter la stratégie pure j_2 telle qu'elle rend minimale sa perte moyenne. En choisissant une stratégie pure dans la partie suivante, le premier joueur part de considérations analogues. La stratégie pure i_3 qu'il a choisie doit être la meilleure par rapport à la stratégie mixte $W^{(2)}$ composée de stratégies pures j_1 et j_2, chacune avec la probabilité 1/2. La stratégie j_3 choisie après cela par le deuxième joueur doit lui assurer la perte moyenne minimale pour la stratégie mixte de l'adversaire, comprenant les stratégies pures i_1, i_2, i_3 avec la probabilité 1/3 chacune. Le perfectionnement successif des stratégies se poursuit aussi longtemps que l'on veut. En dénombrant après chaque partie les fréquences d'apparition des stratégies pures de l'adversaire au cours

de tout le jeu, chacun des joueurs les prend pour les probabilités avec lesquelles les stratégies pures rentrent dans les stratégies mixtes de l'adversaire et choisit une stratégie pure suivante la meilleure par rapport à la stratégie mixte donnée de l'autre joueur.

Le schéma simple « d'information par habitude » ainsi décrit peut être formalisé de la façon suivante.

Supposons qu'après la s-ième partie le premier joueur ait découvert que son adversaire a choisi s_j fois la j-ième stratégie pure ($j = 1, 2, \ldots, n$). Pour cette raison, il adopte l'hypothèse que son adversaire s'en tient à la stratégie mixte $W^{(s)} = \left(\frac{s_1}{s}, \frac{s_2}{s}, \ldots, \frac{s_n}{s}\right)$.

Ceci étant admis, le premier joueur choisit dans la $(s+1)$-ième partie la stratégie pure $e_{i_{s+1}}$ qui donne le gain moyen maximal par rapport à $W^{(s)}$. Le numéro i_{s+1} de la stratégie pure choisie est le numéro de la composante maximale du vecteur $AW^{(s)}$.

D'une manière analogue, le deuxième joueur, après t parties, suppose que la stratégie mixte de l'adversaire ait la forme $U^{(t)} = \left(\frac{t_1}{t}, \ldots, \frac{t_m}{t}\right)$, où t_i est la fréquence d'apparition de la i-ème stratégie dans les parties précédentes. Dans la $(t+1)$-ième partie, le deuxième joueur opte pour la stratégie pure $e_{j_{t+1}}$ qui lui assure la perte moyenne minimale. Le numéro j_{t+1} de cette stratégie pure est le même que celui de la composante minimale du vecteur $U^{(t)}A$.

Il vient

$$U^{(s+1)} = \frac{s}{s+1} U^{(s)} + \frac{1}{s+1} e_{i_{s+1}}, \tag{3.1}$$

$$W^{(t+1)} = \frac{t}{t+1} W^{(t)} + \frac{1}{t+1} e_{j_{t+1}}, \tag{3.2}$$

où i_{s+1} est le numéro de la composante maximale du vecteur $AW^{(s)}$; j_{t+1}, le numéro de la composante minimale du vecteur $U^{(t)}A$.

Dans la description du jeu fictif, nous avons supposé que les deux joueurs fissent leur choix à tour de rôle. Dans la littérature sur la théorie des jeux, on considère le plus souvent la procédure itérative dans laquelle les stratégies des deux joueurs sont déterminées en même temps. Dans cette procédure, les deux joueurs font leur premier choix simultanément et, recevant l'information sur la décision de l'adversaire, adoptent simultanément dans chaque partie consécutive la stratégie pure la plus avantageuse par rapport à la stratégie mixte élaborée jusqu'alors par l'adversaire.

L'expérience accumulée jusqu'à présent dans la résolution des jeux montre que la première procédure (avec choix séquentiel des stratégies) converge beaucoup plus vite que la deuxième (avec choix simultané des stratégies).

J. Robinson [66] a démontré que dans la méthode du jeu fictif le gain, après un grand nombre de parties, se stabilise et tend vers

la valeur du jeu. Le théorème de Robinson sur la convergence de la procédure itérative décrite ci-dessus affirme que

$$\lim_{s\to\infty} \min_j (U^{(s)}A) = \lim_{s\to\infty} \max_i (AW^{(s)}) = v, \tag{3.3}$$

où v est la valeur du jeu.

Nous ne donnerons pas ici la démonstration assez longue de cette proposition intuitivement naturelle. Constatons seulement que la convergence de la procédure itérative est très lente. [81] en donne l'appréciation suivante. L'erreur dans la détermination de la valeur du jeu v après s étapes de la méthode itérative de Brown avec choix simultané des stratégies ne dépasse pas une grandeur de l'ordre de $s^{-\frac{1}{m+n-2}}$.

Plus précisément,

$$|v - \max_i AW^{(s)}| < Cs^{-\frac{1}{m+n-2}}, \tag{3.4}$$

$$|v - \min_j U^{(s)}A| < Cs^{-\frac{1}{m+n-2}}, \tag{3.5}$$

où C = const.

Pour le moment il n'est pas encore clair si cette estimation est précise. Il y a des raisons de considérer son amélioration comme possible. La convergence de la procédure itérative est beaucoup plus rapide si les dimensions de la sous-matrice essentielle sont nettement inférieures à celles de la matrice des gains. Dans [81] est émise une hypothèse sur la possibilité de rendre plus précises les estimations (3.4) et (3.5) en remplaçant les valeurs m et n (nombres de stratégies des deux adversaires) par les nombres $\overline{m}$ et $\overline{n}$ qui déterminent les dimensions de la sous-matrice essentielle de la matrice A.

Pour les calculs relatifs au «jeu fictif», il est commode d'appliquer la disposition représenteé à la page suivante.

La première colonne du tableau indique le numéro de la partie. Les groupes suivants de m et n colonnes (composantes des vecteurs $sU^{(s)}A$ et $sAW^{(s)}$ respectivement) donnent les valeurs des gains multipliées par s, sur lesquels peuvent compter les partenaires dans le cas de tel ou tel choix de stratégies pures de la part de leurs adversaires.

Dans la première ligne ($s = 1$), cases 1, 2, . . ., n des colonnes $sU^{(s)}A$, on porte une ligne arbitraire de la matrice des gains, correspondant au premier choix du premier joueur. On encadre l'élément minimal de la ligne.

Le numéro de la case marquée détermine la colonne de la matrice de paiement, dont les éléments sont portés en première ligne dans les cases du groupe de colonnes $sAW^{(s)}$. On encadre l'élément maximal de la colonne. Le numéro de la case marquée détermine celui de la ligne de la matrice A, dont les éléments doivent être ajoutés aux éléments correspondants de la ligne choisie dans la première

	$sU^{(s)}A$	$sAW^{(s)}$	$U^{(s)}$	$W^{(s)}$	$v_1^{(s)}$	$v_2^{(s)}$	$\delta v^{(s)}$
s \ i, j	1 2 ... n	1 2 ... m	1 2 ... m	1 2 ... n			
1	...	...	...	...			
2	...	...	...	...			
3	...	...	...	...			
⋮	⋮ ⋮	⋮ ⋮	⋮ ⋮	⋮ ⋮	⋮	⋮	⋮

partie. Les sommes sont écrites dans les cases 1, 2, . . ., n de la deuxième ligne ($s = 2$) dans les groupes de colonnes $sU^{(s)}A$. La somme minimale indique le numéro de la colonne de la matrice A, dont les éléments doivent être additionnés aux éléments respectifs de la colonne choisie dans la première partie. Les sommes sont portées dans les cases 1, 2, . . ., m de la deuxième ligne dans les groupes de colonnes $sAW^{(s)}$. On procède d'après les mêmes règles pour remplir les cases des groupes de colonnes $sU^{(s)}A$ et $sAW^{(s)}$ des lignes suivantes.

Après les colonnes déterminant les gains des joueurs dans les parties précédentes, on complète dans le tableau les groupes de m et n colonnes contenant les composantes des stratégies mixtes choisies par les joueurs dans les parties successives.

Les colonnes $v_1^{(s)}$ et $v_2^{(s)}$ indiquent les valeurs du gain assuré du premier joueur et de la perte assurée du deuxième. Les valeurs $v_1^{(s)}$ et $v_2^{(s)}$ sont naturellement égales aux éléments encadrés de la s-ième ligne (dans les cases des groupes de colonnes $sU^{(s)}A$ et $sAW^{(s)}$ respectivement), divisés par s. La dernière colonne contient les valeurs $\delta v^{(s)} = 2\,\frac{v_2^{(s)} - v_1^{(s)}}{v_1^{(s)} + v_2^{(s)}}$. (On suppose que la matrice non négative des gains compte dans chaque colonne au moins un élément positif.) La valeur $\delta v^{(s)}$ caractérise l'erreur relative de détermination de la valeur du jeu après la s-ième partie.

Les tableaux 2.3 et 2.4 donnent les 36 premières étapes (itérations) de la résolution du jeu à matrice de paiement

$$A = \begin{Vmatrix} 1 & 2 & 3 \\ 4 & 0 & 1 \\ 2 & 3 & 0 \end{Vmatrix}.$$

Tableau 2.3

s	$sU^{(s)}A$			$sAW^{(s)}$			$U^{(s)}$			$W^{(s)}$			$v_1^{(s)}$	$v_2^{(s)}$	$\delta v^{(s)}$
	1	2	3	1	2	3	1	2	3	1	2	3			
1	$\boxed{1}$	2	3	1	$\boxed{4}$	2	1	0	0	1	0	0	1	4	1,2
2	5	$\boxed{2}$	4	2	$\boxed{8}$	4	0,50	0,50	0	1	0	0	1	4	1,2
3	9	$\boxed{2}$	5	4	$\boxed{8}$	7	0,33	0,67	0	0,67	0,33	0	0,67	2,67	1,2
4	13	$\boxed{2}$	6	6	8	$\boxed{10}$	0,25	0,75	0	0,50	0,50	0	0,50	2,50	1,33
5	15	$\boxed{5}$	6	8	8	$\boxed{13}$	0,20	0,60	0,20	0,40	0,60	0	1	2,60	0,89
6	17	8	$\boxed{6}$	10	8	$\boxed{16}$	0,17	0,50	0,33	0,33	0,67	0	1	2,67	0,71
7	19	11	$\boxed{6}$	13	9	$\boxed{16}$	0,14	0,43	0,43	0,29	0,57	0,14	0,86	2,29	0,91
8	21	14	$\boxed{6}$	$\boxed{16}$	10	16	0,12	0,38	0,50	0,25	0,50	0,25	0,75	2	0,91
9	22	16	$\boxed{9}$	$\boxed{19}$	11	16	0,22	0,33	0,44	0,22	0,44	0,33	1	2,11	0,71
10	23	18	$\boxed{12}$	$\boxed{22}$	12	16	0,30	0,30	0,40	0,20	0,40	0,40	1,20	2,20	0,59
11	24	20	$\boxed{15}$	$\boxed{25}$	13	16	0,36	0,27	0,36	0,18	0,36	0,45	1,36	2,27	0,50
12	25	22	$\boxed{18}$	$\boxed{28}$	14	16	0,42	0,25	0,33	0,17	0,33	0,50	1,50	2,33	0,44
13	26	24	$\boxed{21}$	$\boxed{31}$	15	16	0,46	0,23	0,31	0,15	0,31	0,54	1,62	2,39	0,38
14	27	26	$\boxed{24}$	$\boxed{34}$	16	16	0,50	0,21	0,29	0,14	0,29	0,57	1,71	2,43	0,34
15	28	28	$\boxed{27}$	$\boxed{37}$	17	16	0,53	0,20	0,27	0,13	0,27	0,60	1,80	2,47	0,31
16	$\boxed{29}$	30	30	$\boxed{40}$	18	16	0,56	0,19	0,25	0,12	0,25	0,62	1,81	2,50	0,32

17	$\boxed{30}$	32	33	$\boxed{41}$	22	18	0,59	0,18	0,23	0,18	0,23	0,59	1,76	2,41	0,36
18	$\boxed{31}$	34	36	$\boxed{42}$	26	20	0,61	0,17	0,22	0,22	0,22	0,56	1,72	2,33	0,30
19	$\boxed{32}$	36	39	$\boxed{43}$	30	22	0,63	0,16	0,21	0,26	0,21	0,53	1,68	2,26	0,29
20	$\boxed{33}$	38	42	$\boxed{44}$	34	24	0,65	0,15	0,20	0,30	0,20	0,50	1,65	2,20	0,29
21	$\boxed{34}$	40	45	$\boxed{45}$	38	26	0,67	0,14	0,19	0,33	0,19	0,48	1,62	2,14	0,28
22	$\boxed{35}$	42	48	$\boxed{46}$	42	28	0,68	0,14	0,18	0,37	0,18	0,45	1,59	2,09	0,27
23	$\boxed{36}$	44	51	$\boxed{47}$	46	30	0,70	0,13	0,17	0,39	0,17	0,44	1,57	2,04	0,27
24	$\boxed{37}$	46	54	48	$\boxed{50}$	32	0,71	0,12	0,17	0,42	0,17	0,42	1,54	2,08	0,30
25	$\boxed{41}$	46	55	49	$\boxed{54}$	34	0,68	0,16	0,16	0,44	0,16	0,40	1,64	2,16	0,27
26	$\boxed{45}$	46	56	50	$\boxed{58}$	36	0,65	0,19	0,15	0,46	0,15	0,38	1,73	2,23	0,25
27	49	$\boxed{46}$	57	51	$\boxed{62}$	38	0,63	0,22	0,15	0,48	0,15	0,37	1,70	2,30	0,30
28	53	$\boxed{46}$	58	53	$\boxed{62}$	41	0,61	0,25	0,14	0,46	0,18	0,36	1,64	2,21	0,30
29	57	$\boxed{46}$	59	55	$\boxed{62}$	44	0,59	0,28	0,14	0,45	0,21	0,34	1,59	2,14	0,30
30	61	$\boxed{46}$	60	57	$\boxed{62}$	47	0,57	0,30	0,13	0,43	0,23	0,33	1,53	2,07	0,30
31	65	$\boxed{46}$	61	59	$\boxed{62}$	50	0,55	0,32	0,13	0,42	0,26	0,32	1,48	2,00	0,30
32	69	$\boxed{46}$	62	61	$\boxed{62}$	53	0,53	0,34	0,13	0,41	0,28	0,31	1,44	1,94	0,30
33	73	$\boxed{46}$	63	$\boxed{63}$	62	56	0,52	0,36	0,12	0,39	0,30	0,30	1,39	1,91	0,31
34	74	$\boxed{48}$	66	$\boxed{65}$	62	59	0,53	0,35	0,12	0,38	0,32	0,29	1,41	1,91	0,30
35	75	$\boxed{50}$	69	$\boxed{67}$	62	62	0,54	0,34	0,11	0,37	0,34	0,29	1,43	1,91	0,29
36	76	$\boxed{52}$	72	$\boxed{69}$	62	65	0,56	0,33	0,11	0,36	0,36	0,28	1,44	1,92	0,28

Tableau 2.4

	$sU^{(s)}A$			$sAW^{(s)}$			$U^{(s)}$			$W^{(s)}$			$v_1^{(s)}$	$v_2^{(s)}$	$\delta v^{(s)}$
s	1	2	3	1	2	3	1	2	3	1	2	3			
1	$\boxed{1}$	2	3	1	$\boxed{4}$	2	1	0	0	1	0	0	1	4	1,2
2	5	$\boxed{2}$	4	3	4	$\boxed{5}$	0,50	0,50	0	0,50	0,50	0	1	2,5	0,86
3	7	5	$\boxed{4}$	$\boxed{6}$	5	5	0,33	0,33	0,33	0,33	0,33	0,33	1,33	2	0,4
4	8	$\boxed{7}$	7	$\boxed{8}$	5	8	0,50	0,25	0,25	0,25	0,50	0,25	1,75	2	0,13
5	$\boxed{9}$	9	10	9	9	$\boxed{10}$	0,60	0,20	0,20	0,40	0,40	0,20	1,80	2	0,11
6	11	12	$\boxed{10}$	$\boxed{12}$	10	10	0,50	0,17	0,33	0,33	0,33	0,33	1,67	2	0,18
7	$\boxed{12}$	14	13	13	$\boxed{14}$	12	0,57	0,14	0,29	0,43	0,29	0,29	1,72	2	0,15
8	16	$\boxed{14}$	14	$\boxed{15}$	14	15	0,50	0,25	0,25	0,38	0,38	0,25	1,75	1,88	0,07
9	17	$\boxed{16}$	17	17	14	$\boxed{18}$	0,56	0,22	0,22	0,33	0,44	0,22	1,78	2	0,12
10	19	19	$\boxed{17}$	$\boxed{20}$	15	18	0,50	0,20	0,30	0,30	0,40	0,30	1,70	2	0,16
11	$\boxed{20}$	21	20	$\boxed{21}$	19	20	0,55	0,18	0,27	0,36	0,36	0,27	1,82	1,91	0,05
12	$\boxed{21}$	23	23	22	$\boxed{23}$	22	0,58	0,17	0,25	0,42	0,33	0,25	1,75	1,92	0,09
13	25	$\boxed{23}$	24	24	23	$\boxed{25}$	0,54	0,23	0,23	0,38	0,38	0,23	1,77	1,92	0,08
14	27	26	$\boxed{24}$	$\boxed{27}$	24	25	0,50	0,21	0,29	0,36	0,36	0,29	1,71	1,93	0,12
15	28	28	$\boxed{27}$	$\boxed{30}$	25	25	0,53	0,20	0,27	0,33	0,33	0,33	1,80	2	0,10
16	$\boxed{29}$	30	30	$\boxed{31}$	29	27	0,56	0,19	0,25	0,38	0,31	0,31	1,81	1,94	0,07

17	$\boxed{30}$	32	33	32	$\boxed{33}$	29	0,59	0,18	0,24	0,41	0,29	0,29	1,76	1,94	0,10
18	34	$\boxed{32}$	34	$\boxed{34}$	33	32	0,56	0,22	0,22	0,39	0,33	0,28	1,78	1,89	0,06
19	35	$\boxed{34}$	37	$\boxed{36}$	33	35	0,58	0,21	0,21	0,37	0,37	0,26	1,79	1,90	0,06
20	$\boxed{36}$	36	40	$\boxed{37}$	37	37	0,60	0,20	0,20	0,40	0,35	0,25	1,80	1,85	0,03
21	$\boxed{37}$	38	43	38	$\boxed{41}$	39	0,62	0,19	0,19	0,43	0,33	0,24	1,76	1,95	0,10
22	41	$\boxed{38}$	44	40	41	$\boxed{42}$	0,59	0,23	0,18	0,41	0,36	0,23	1,73	1,91	0,10
23	43	$\boxed{41}$	44	42	41	$\boxed{45}$	0,56	0,22	0,22	0,39	0,39	0,22	1,78	1,96	0,09
24	45	$\boxed{44}$	44	44	41	$\boxed{48}$	0,54	0,21	0,25	0,38	0,42	0,21	1,83	2	0,09
25	47	47	$\boxed{44}$	47	42	$\boxed{48}$	0,52	0,20	0,28	0,36	0,40	0,24	1,76	1,92	0,09
26	49	50	$\boxed{44}$	$\boxed{50}$	43	48	0,50	0,19	0,31	0,35	0,38	0,27	1,69	1,92	0,13
27	50	52	$\boxed{47}$	$\boxed{53}$	44	48	0,52	0,18	0,30	0,33	0,37	0,30	1,74	1,96	0,12
28	51	54	$\boxed{50}$	$\boxed{56}$	45	48	0,54	0,18	0,29	0,32	0,36	0,32	1,79	2	0,11
29	$\boxed{52}$	56	53	$\boxed{57}$	49	50	0,55	0,17	0,28	0,34	0,34	0,31	1,79	1,97	0,09
30	$\boxed{53}$	58	56	$\boxed{58}$	53	52	0,57	0,17	0,27	0,37	0,33	0,30	1,77	1,93	0,09
31	$\boxed{54}$	60	59	$\boxed{59}$	57	54	0,58	0,16	0,26	0,39	0,32	0,29	1,74	1,90	0,09
32	$\boxed{55}$	62	62	60	$\boxed{61}$	56	0,59	0,16	0,25	0,41	0,31	0,28	1,72	1,91	0,10
33	$\boxed{59}$	62	63	61	$\boxed{65}$	58	0,58	0,18	0,24	0,42	0,30	0,27	1,79	1,97	0,10
34	63	$\boxed{62}$	64	63	$\boxed{65}$	61	0,56	0,21	0,24	0,41	0,32	0,26	1,82	1,91	0,05
35	67	$\boxed{62}$	65	$\boxed{65}$	65	64	0,54	0,23	0,23	0,40	0,34	0,26	1,77	1,86	0,05
36	68	$\boxed{64}$	68	$\boxed{67}$	65	67	0,56	0,22	0,22	0,39	0,36	0,25	1,78	1,86	0,05

La solution exacte de ce jeu est obtenue au § 3.1 à l'aide de la méthode du simplexe.

Le tableau 2.3 correspond à la méthode du jeu fictif avec choix simultané des stratégies. Le tableau 2.4 répond à un choix séquentiel.

Comme il ressort des tableaux (surtout dans le cas du choix séquentiel), quand le numéro de l'itération augmente la variation de $v_1^{(s)}$ et $v_2^{(s)}$ n'est pas monotone. En interrompant la procédure après

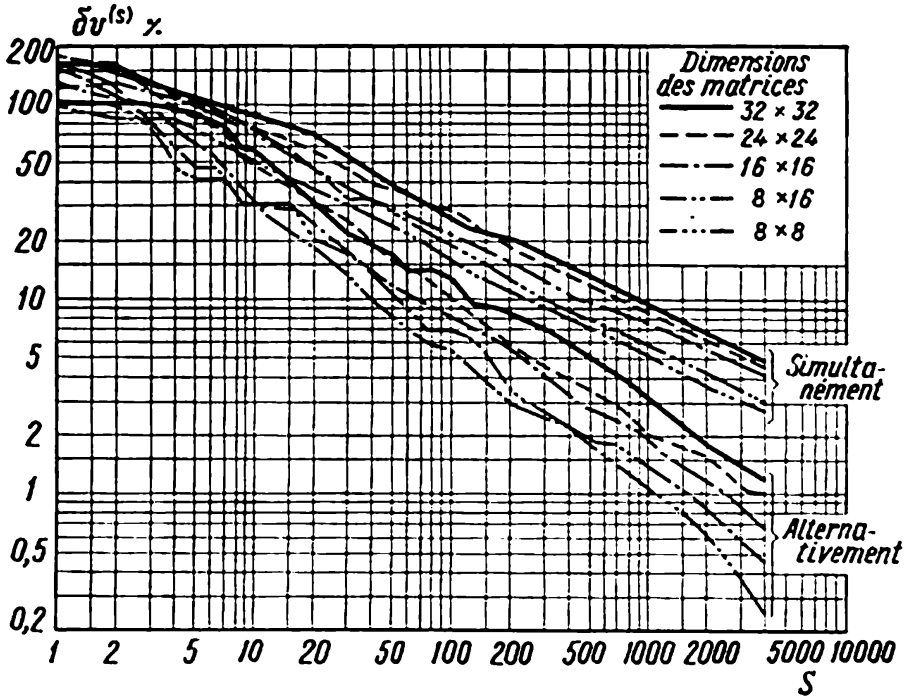

Fig. 2.1

le nombre d'itérations prévu d'avance, il est rationnel de choisir la stratégie du premier joueur d'après le maximum de $v_1^{(s)}$, et celle du deuxième d'après le minimum de $v_2^{(s)}$.

Les tableaux 2.5 et 2.6 donnent les valeurs de $\delta v^{(s)}$ en fonction du numéro de l'itération respectivement pour les cas du choix simultané et du choix séquentiel des stratégies. On voit que le choix séquentiel des stratégies assure δv imposée en un nombre d'itérations plus petit.

La figure 2.1 donne les courbes de variation de $\delta v^{(s)}$ en fonction de s calculées lors de la détermination des stratégies optimales par la méthode du jeu fictif avec choix simultané et séquentiel des stratégies, pour des jeux à matrices de paiement de diverses dimensions. Les éléments des matrices de paiement ont été choisis de façon aléatoire. Les dimensions des matrices sont indiquées sur la figure.

Comme on le voit, la procédure itérative converge dans tous les cas beaucoup plus vite qu'on pouvait s'y attendre d'après les formules d'estimation (3.4) et (3.5).

Tableau 2.5

s	$\delta v^{(s)}$	s	$\delta v^{(s)}$	s	$\delta v^{(s)}$
5	0,889	340	0,070	680	0,052
10	0,589	350	0,070	690	0,053
20	0,286	360	0,068	700	0,052
30	0,300	370	0,067	710	0,051
40	0,189	380	0,066	720	0,051
50	0,151	390	0,064	730	0,050
60	0,162	400	0,033	740	0,049
70	0,155	410	0,061	750	0,049
80	0,156	420	0,060	760	0,048
90	0,138	430	0,058	770	0,047
100	0,124	440	0,057	780	0,046
110	0,113	450	0,056	790	0,046
120	0,117	460	0,058	800	0,045
130	0,112	470	0,060	810	0,045
140	0,108	480	0,063	820	0,045
150	0,101	490	0,061	830	0,044
160	0,098	500	0,060	840	0,046
170	0,092	510	0,060	850	0,046
180	0,087	520	0,059	860	0,046
190	0,083	530	0,058	870	0,046
200	0,079	540	0,057	880	0,045
210	0,085	550	0,056	890	0,045
220	0,082	560	0,056	900	0,046
230	0,083	570	0,055	910	0,045
240	0,084	580	0,054	920	0,046
250	0,085	590	0,053	930	0,045
260	0,082	600	0,054	940	0,045
270	0,078	610	0,054	950	0,044
280	0,076	620	0,055	960	0,044
290	0,073	630	0,054	970	0,043
300	0,071	640	0,054	980	0,043
310	0,072	650	0,054	990	0,043
320	0,073	660	0,054	1000	0,042
330	0,072	670	0,053	1010	0,042

Tableau 2.6

s	$\delta v^{(s)}$	s	$\delta v^{(s)}$	s	$\delta v^{(s)}$
5	0,108	340	0,019	680	0,0064
10	0,162	350	0,020	690	0,0063
20	0,027	360	0,017	700	0,0062
30	0,090	370	0,013	710	0,0069
40	0,014	380	0,010	720	0,0075
50	0,011	390	0,014	730	0,0044
60	0,027	400	0,011	740	0,0029
70	0,046	410	0,012	750	0,0036
80	0,041	420	0,009	760	0,0028
90	0,048	430	0,010	770	0,0054
100	0,038	440	0,012	780	0,0035
110	0,034	450	0,012	790	0,0062
120	0,032	460	0,011	800	0,0054
130	0,033	470	0,012	810	0,0033
140	0,042	480	0,009	820	0,0033
150	0,032	490	0,012	830	0,0026
160	0,034	500	0,011	840	0,0026
170	0,029	510	0,012	850	0,0032
180	0,021	520	0,012	860	0,0031
190	0,020	530	0,011	870	0,0025
200	0,019	540	0,010	880	0,0012
210	0,015	550	0,011	890	0,0012
220	0,017	560	0,010	900	0,0030
230	0,026	570	0,0085	910	0,0042
240	0,022	580	0,0075	920	0,0047
250	0,015	590	0,0100	930	0,0046
260	0,019	600	0,0081	940	0,0069
270	0,018	610	0,0089	950	0,0063
280	0,012	620	0,0078	960	0,0062
290	0,011	630	0,0060	970	0,0072
300	0,014	640	0,0051	980	0,0077
310	0,016	650	0,0050	990	0,0065
320	0,019	660	0,0057	1000	0,0070
330	0,015	670	0,0056	1010	0,0064

3.3. Considérons la méthode itérative de résolution numérique des jeux rectangulaires proposée par G. von Neumann [59].

Les raisonnements cités dans le paragraphe précédent entraînent que la résolution d'un jeu à matrice de paiement $\| a_{ij} \|_{m \cdot n}$ se ramène au choix de nombres u_i $(i = 1, 2, \ldots, m)$, w_j $(j = 1, 2, \ldots, n)$ et v tels que:

$$\sum_{i=1}^{m} a_{ij} u_i \geqslant v, \quad j = 1, 2, \ldots, n; \tag{3.6}$$

$$\sum_{j=1}^{n} a_{ij} w_j \leqslant v, \quad i = 1, 2, \ldots, m; \tag{3.7}$$

$$u_i \geqslant 0, \quad i = 1, 2, \ldots, m; \tag{3.8}$$

$$\sum_{i=1}^{m} u_i = 1; \tag{3.9}$$

$$w_j \geqslant 0, \quad j = 1, 2, \ldots, n; \tag{3.10}$$

$$\sum_{j=1}^{n} w_j = 1. \tag{3.11}$$

Introduisons dans la discussion les variables d'écart

$$P_i = \sum_{j=1}^{n} a_{ij} w_j - v, \tag{3.12}$$

$$Q_j = v - \sum_{i=1}^{m} a_{ij} u_i, \tag{3.13}$$

qui déterminent à leur tour les paramètres non négatifs

$$p_i = \max (P_i, 0), \tag{3.14}$$

$$q_j = \max (Q_j, 0). \tag{3.15}$$

Les valeurs de la fonction

$$\Phi = \sum_{i=1}^{m} p_i^2 + \sum_{j=1}^{n} q_j^2 \tag{3.16}$$

donnent une estimation de la précision avec laquelle les stratégies retenues $U = (u_1, \ldots, u_m)$ et $W = (w_1, \ldots, w_n)$ et la quantité v approchent respectivement des stratégies oprimales et de la valeur du jeu. Pour le vecteur (U, W, v) qui représente la solution du jeu, la fonction Φ devient nulle.

La méthode de Neumann est une procédure itérative de calcul séquentiel de vecteurs U et W et du scalaire v qui font tendre Φ vers zéro.

Les vecteurs initiaux U^0 et W^0 peuvent être choisis parmi des vecteurs quelconques vérifiant les conditions (3.8)-(3.11). On pose

généralement

$$u_i^0 = \frac{1}{m}, \quad w_j^0 = \frac{1}{n}, \quad v^0 = \frac{1}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} a_{ij}. \tag{3.17}$$

Le passage de la s-ième à la $(s+1)$-ième itération s'effectue selon les formules:

$$u_i^{(s+1)} = (1 - \alpha^{(s)}) u_i^{(s)} + \alpha^{(s)} \overline{u}_i^{(s)}, \tag{3.18}$$

$$w_j^{(s+1)} = (1 - \alpha^{(s)}) w_j^{(s)} + \alpha^{(s)} \overline{w}_j^{(s)}, \tag{3.19}$$

$$v^{(s+1)} = (1 - \alpha^{(s)}) v^{(s)} + \alpha^{(s)} \overline{v}^{(s)}, \tag{3.20}$$

où les corrections $\overline{u}_i^{(s)}$ et $\overline{w}_j^{(s)}$ des stratégies et la correction $\overline{v}^{(s)}$ de la grandeur $v^{(s)}$ se calculent d'après les formules:

$$\overline{u}_i^{(s)} = \frac{p_i^{(s)}}{\sum_{i=1}^{m} p_i^{(s)}}, \quad \overline{w}_j^{(s)} = \frac{q_j^{(s)}}{\sum_{j=1}^{n} q_j^{(s)}}, \quad \overline{v}^{(s)} = \sum_{j=1}^{n} \sum_{i=1}^{m} a_{ij} \overline{u}_i^{(s)} \overline{w}_j^{(s)}. \tag{3.21}$$

Si $\sum_{i=1}^{m} p_i^{(s)}$ (ou $\sum_{j=1}^{n} q_j^{(s)}$) est nulle, on peut choisir pour $\overline{u}_i^{(s)}$ (respectivement pour $\overline{w}_j^{(s)}$) n'importe quels nombres non négatifs dont la somme est égale à l'unité.

Les relations

$$\sum_{i=1}^{m} p_i^{(s)} = \sum_{j=1}^{n} q_j^{(s)} = 0$$

attestent que la solution est trouvée et qu'il faut mettre fin à la procédure.

La grandeur $\alpha^{(s)}$ est un paramètre de commande qui détermine selon les formules (3.18)-(3.20) le passage d'une itération à l'itération suivante. Le choix de $\alpha^{(s)}$ doit assurer la décroissance de la fonction Φ d'une itération à l'autre.

Comparons les valeurs de Φ pour deux itérations voisines. Il est facile de vérifier directement que les variables d'écart $(P_i^{(s+1)}, Q_j^{(s+1)})$ et $(P_i^{(s)}, Q_j^{(s)})$ dans des itérations voisines sont liées entre elles par des formules du type (3.18)-(3.20):

$$P_i^{(s+1)} = (1 - \alpha^{(s)}) P_i^{(s)} + \alpha^{(s)} \overline{P}_i^{(s)},$$

$$Q_j^{(s+1)} = (1 - \alpha^{(s)}) Q_j^{(s)} + \alpha^{(s)} \overline{Q}_j^{(s)},$$

où

$$\overline{P}_i^{(s)} = \sum_{j=1}^{n} a_{ij} \overline{w}_j^{(s)} - \overline{v}^{(s)}, \tag{3.22}$$

$$\overline{Q}_j^{(s)} = \overline{v}^{(s)} - \sum_{i=1}^{m} a_{ij} \overline{u}_i^{(s)}. \tag{3.23}$$

D'après les formules (3.14), $p_i = 0$ avec $P_i \leqslant 0$ et $p_i = P_i$ avec $P_i > 0$. C'est pourquoi on a toujours

$$[p_i^{(s+1)}]^2 \leqslant [(1 - \alpha^{(s)})p_i^{(s)} + \alpha^{(s)}\overline{P}_i^{(s)}]^2,$$

et de façon analogue,

$$[q_j^{(s+1)}]^2 \leqslant [(1 - \alpha^{(s)})q_j^{(s)} + \alpha^{(s)}\overline{Q}_j^{(s)}]^2.$$

Les inégalités obtenues permettent de comparer les valeurs de $\Phi^{(s+1)}$ et $\Phi^{(s)}$. On a

$$\Phi^{(s+1)} = \sum_{i=1}^{m} [p_i^{(s+1)}]^2 + \sum_{j=1}^{n} [q_j^{(s+1)}]^2 \leqslant [1-\alpha^{(s)}]^2 \Big(\sum_{i=1}^{m} [p_i^{(s)}]^2 + \\ + \sum_{j=1}^{n} [q_j^{(s)}]^2\Big) + 2\alpha^{(s)}(1-\alpha^{(s)})\Big(\sum_{i=1}^{m} p_i^s \overline{P}_i^{(s)} + \\ + \sum_{j=1}^{n} q_j^{(s)}\overline{Q}_j^{(s)}\Big) + [\alpha^{(s)}]^2\Big(\sum_{i=1}^{m} [\overline{P}_i^{(s)}]^2 + \sum_{j=1}^{n} [\overline{Q}_j^{(s)}]^2\Big).$$

Les formules (3.21)-(3.23) amènent

$$\sum_{i=1}^{m} p_i^{(s)}\overline{P}_i^{(s)} + \sum_{j=1}^{n} q_j^{(s)}\overline{Q}_j^{(s)} = 0.$$

Par conséquent,

$$\Phi^{(s+1)} \leqslant [1-\alpha^{(s)}]^2\Phi^{(s)} + [\alpha^{(s)}]^2 C^{(s)}, \tag{3.24}$$

où

$$C^{(s)} = \sum_{i=1}^{m} [\overline{P}_i^{(s)}]^2 + \sum_{j=1}^{n} [\overline{Q}_j^{(s)}]^2. \tag{3.25}$$

Le deuxième membre de l'inégalité (3.24) atteint une valeur minimale avec

$$\alpha^{(s)} = \frac{\Phi^{(s)}}{\Phi^{(s)} + C^{(s)}}. \tag{3.26}$$

Constatons que $\Phi^{(s)} \geqslant 0$ et $C^{(s)} \geqslant 0$. C'est pourquoi $0 \leqslant \alpha^{(s)} \leqslant 1$.

Lorsque $\alpha^{(s)}$ provient de (3.26), l'inégalité (3.24) se met sous la forme

$$\Phi^{(s+1)} \leqslant \frac{\Phi^{(s)}C^{(s)}}{\Phi^{(s)} + C^{(s)}}$$

ou

$$\frac{1}{\Phi^{(s+1)}} \geqslant \frac{1}{\Phi^{(s)}} + \frac{1}{C^{(s)}}. \tag{3.27}$$

Evaluons la convergence de la méthode de Neumann. On a la proposition suivante.

Théorème 3.1. *La méthode itérative de Neumann est convergente. De plus,* $\Phi^{(s)}$ *tend vers zéro comme* $\frac{1}{s}$.

Démonstration. Soient

$$\lambda = \max_{i,j} a_{ij}, \quad \mu = \min_{i,j} a_{ij}.$$

Il est clair que

$$\lambda \geqslant \sum_{j=1}^{n} a_{ij}\overline{w}_j^{(s)} \geqslant \mu,$$

$$\lambda \geqslant \sum_{i=1}^{m} a_{ij}\overline{u}_i^{(s)} \geqslant \mu,$$

$$\lambda \geqslant \overline{v}^{(s)} \geqslant \mu.$$

On en tire

$$\lambda - \mu \geqslant \sum_{j=1}^{n} a_{ij}\overline{w}_j^{(s)} - \overline{v}^{(s)} = \overline{P}_i^{(s)} \geqslant \mu - \lambda,$$

$$\lambda - \mu \geqslant -\sum_{i=1}^{m} a_{ij}\overline{u}_i^{(s)} + \overline{v}^{(s)} = \overline{Q}_j^{(s)} \geqslant \mu - \lambda,$$

ou

$$\lambda - \mu \geqslant |\overline{Q}_j^{(s)}|,$$

$$\lambda - \mu \geqslant |\overline{P}_i^{(s)}|.$$

Par conséquent (cf. 3.25),

$$C^{(s)} = \sum_{i=1}^{m} [\overline{P}_i^{(s)}]^2 + \sum_{j=1}^{n} [\overline{Q}_j^{(s)}]^2 \leqslant (m+n)(\lambda-\mu)^2.$$

Sous cette condition, l'inégalité (3.27) prend la forme

$$\frac{1}{\Phi^{(s+1)}} \geqslant \frac{1}{\Phi^{(s)}} + \frac{1}{(m+n)(\lambda-\mu)^2}.$$

Le système d'inégalités récurrentes obtenu conduit à la relation

$$\frac{1}{\Phi^{(s+1)}} \geqslant \frac{1}{\Phi^{(0)}} + \frac{s+1}{(m+n)(\lambda-\mu)^2} > \frac{s+1}{(m+n)(\lambda-\mu)^2}.$$

D'où

$$\Phi^{(s)} < \frac{a}{s},$$

où $a = (m+n)(\lambda-\mu)^2$, ce qu'il fallait démontrer.

Le calcul d'après la méthode de von Neumann ne présente aucune difficulté. Le schéma bloc des calculs est représenté fig. 2.2 Voici les notations adoptées :

$$p = \sum_{i=1}^{m} p_i, \quad q = \sum_{j=1}^{n} q_j,$$

$$S_j = \sum_{i=1}^{m} a_{ij}\overline{u}_i, \quad T_i = \sum_{j=1}^{n} a_{ij}\overline{w}.$$

Le tableau 2.7 donne les paramètres des 12 premières itérations de la résolution du jeu à matrice de paiement

$$A = \begin{Vmatrix} 1 & 2 & 3 \\ 4 & 0 & 1 \\ 2 & 3 & 0 \end{Vmatrix}$$

par la méthode de Neumann. Ce jeu a été résolu au § 3.1 par la méthode du simplexe et au § 3.2 par les différentes variantes de la méthode du jeu fictif. Le tableau 2.8 consigne les valeurs de $\delta v^{(s)}$ et $\Phi^{(s)}$ en fonction du numéro de l'itération. La comparaison du tableau 2.8 avec les tableaux 2.5 et 2.6 montre que malgré une évaluation de la convergence de la méthode de Neumann plus favorable que l'évaluation théorique de la méthode du jeu fictif, le choix séquentiel des stratégies de cette dernière méthode assure plus rapidement la précision imposée à la solution du problème. La résolution de plusieurs autres exemples a conduit à des résultats analogues. Néanmoins, l'analyse d'exemples isolés ne peut encore servir d'argument de poids pour comparer ces méthodes.

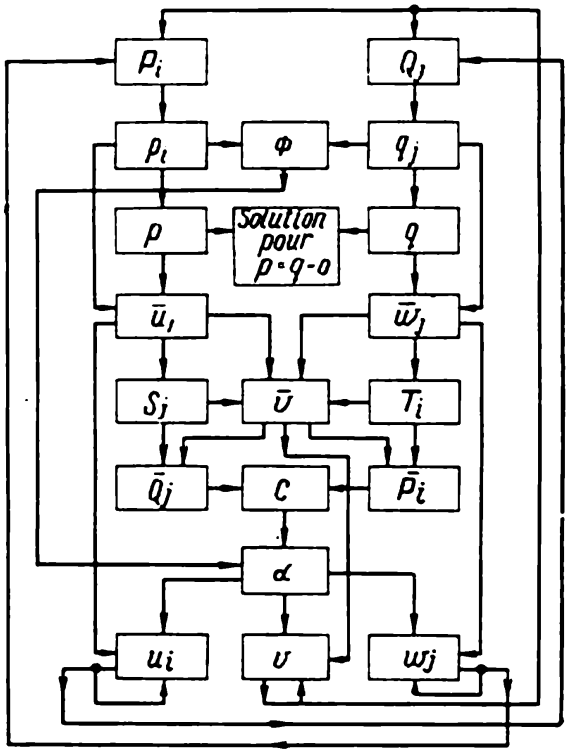

Fig. 2.2

D'après le schéma bloc représenté figure 2.2, on peut évaluer le nombre d'opérations à chaque étape de la méthode. On a :

nombre de comparaisons (calcul de max $(0, P_i)$ et max $(0, Q_j)$) : $(m + n)$;

nombre d'additions et de soustractions : $[2mn + 7(m + n) + (m \text{ ou } n) + 4]$;

nombre de multiplications : $[2mn + 4(m + n) + (m \text{ ou } n) + 1]$;

nombre de divisions : $(m + n + 1)$.

3.4. Les étapes isolées des méthodes itératives ne présentent généralement pas de grandes difficultés. Cependant, la convergence de ces méthodes est très lente. C'est pourquoi, lorsqu'il faut obtenir une solution relativement précise d'un jeu ou d'un problème de programmation linéaire, il est préférable d'opter pour les méthodes finies.

Tableau 2.7

s \ i, j	$\Phi^{(s)}$	$\alpha^{(s)}$	$u_i^{(s)}$			$w_j^{(s)}$			$P_i^{(s)}$			$Q_j^{(s)}$			$v^{(s)}$	$v_1^{(s)}$	$v_2^{(s)}$	$\delta v^{(s)}$
	—	—	1	2	3	1	2	3	1	2	3	1	2	3	—	—	—	—
0	—	—	0,333	0,333	0,333	0,333	0,333	0,333	0,222	—0,111	—0,111	—0,556	0,111	0,444	1,778	1,333	2,000	0,400
1	0,259	0,0199	0,347	0,327	0,327	0,327	0,331	0,343	0,218	—0,149	—0,153	—0,509	0,125	0,432	1,775	1,367	2,016	0,384
2	0,249	0,0200	0,360	0,320	0,320	0,320	0,329	0,351	0,213	—0,186	—0,192	—0,463	0,138	0,418	1,774	1,399	2,031	0,369
3	0,240	0,0200	0,372	0,314	0,314	0,314	0,327	0,359	0,209	—0,222	—0,228	—0,419	0,150	0,405	1,773	1,431	2,046	0,353
4	0,230	0,0200	0,385	0,307	0,307	0,307	0,326	0,367	0,205	—0,258	—0,262	—0,376	0,162	0,392	1,773	1,463	2,059	0,339
5	0,221	0,0200	0,397	0,301	0,301	0,301	0,325	0,374	0,201	—0,292	—0,293	—0,334	0,173	0,378	1,775	1,493	2,072	0,324
6	0,213	0,0200	0,409	0,295	0,295	0,295	0,325	0,380	0,197	—0,327	—0,322	—0,294	0,183	0,364	1,777	1,523	2,084	0,311
7	0,205	0,0199	0,421	0,289	0,289	0,289	0,325	0,385	0,193	—0,360	—0,349	—0,255	0,192	0,350	1,779	1,553	2,096	0,298
8	0,197	0,0198	0,433	0,284	0,284	0,284	0,326	0,391	0,189	—0,392	—0,373	—0,217	0,201	0,336	1,782	1,581	2,107	0,285
9	0,189	0,0197	0,444	0,278	0,278	0,278	0,327	0,395	0,185	—0,424	—0,396	—0,181	0,210	0,322	1,786	1,609	2,117	0,272
10	0,182	0,0196	0,455	0,273	0,273	0,273	0,328	0,399	0,182	—0,455	—0,416	—0,146	0,218	0,308	1,790	1,637	2,127	0,260
11	0,175	0,0195	0,465	0,267	0,267	0,267	0,330	0,403	0,178	—0,485	—0,434	—0,112	0,225	0,294	1,795	1,663	2,136	0,249
12	0,169	0,0193	0,476	0,262	0,262	0,262	0,332	0,406	0,175	—0,514	—0,450	—0,080	0,231	0,280	1,799	1,689	2,144	0,238

Tableau 2.8

s	$\Phi^{(s)}$	$\delta v^{(s)}$	s	$\Phi^{(s)}$	$\delta v^{(s)}$
10	0,1823	0,260	710	0,0050	0,050
20	0,1220	0,192	760	0,0048	0,047
40	0,0622	0,141	800	0,0047	0,050
60	0,0478	0,138	840	0,0046	0,048
80	0,0413	0,142	880	0,0044	0,042
100	0,0366	0,116	930	0,0042	0,047
120	0,0314	0,117	990	0,0041	0,046
140	0,0270	0,092	1040	0,0040	0,043
160	0,0229	0,090	1080	0,0039	0,042
180	0,0183	0,090	1140	0,0037	0,041
200	0,0160	0,077	1190	0,0036	0,036
220	0,0138	0,081	1230	0,0035	0,035
240	0,0128	0,075	1260	0,0034	0,035
260	0,0118	0,076	1300	0,0032	0,039
280	0,0108	0,070	1350	0,0030	0,035
300	0,0104	0,069	1390	0,0028	0,038
320	0,0099	0,070	1440	0,0027	0,033
340	0,0096	0,063	1480	0,0026	0,036
360	0,0093	0,060	1530	0,0025	0,032
380	0,0090	0,061	1570	0,0024	0,033
400	0,0084	0,054	1640	0,0023	0,031
420	0,0080	0,059	1750	0,0022	0,028
440	0,0077	0,059	1850	0,0020	0,031
460	0,0075	0,053	1980	0,0019	0,030
480	0,0073	0,057	2120	0,0018	0,030
500	0,0071	0,059	2250	0,0017	0,028
520	0,0069	0,058	2360	0,0016	0,027
540	0,0066	0,055	2520	0,0015	0,026
560	0,0063	0,043	2850	0,0014	0,025
580	0,0058	0,054	3330	0,0013	0,025
600	0,0056	0,053	3700	0,0012	0,024
630	0,0054	0,051	4350	0,0011	0,024
650	0,0052	0,050	4780	0,0010	0,021
670	0,0051	0,049	5600	0,0009	0,021

On connaît diverses tentatives d'accélération de la convergence des méthodes itératives. On conçoit intuitivement que l'on puisse accélérer la convergence de la méthode du jeu fictif si, après un certain nombre d'itérations, l'on refuse de prendre en compte l'expérience devenue caduque et l'on ne détermine la stratégie suivante qu'à partir des derniers choix de l'adversaire, le nombre de ces choix étant fixé. Toutefois, l'analyse de cette modification montre que la méthode du jeu fictif à mémoire finie ne garantit pas la convergence de la procédure itérative. Il est possible de construire des exemples où la limitation de la mémoire conduit à une instabilité de la procédure. Une autre modification intuitivement convenable de la méthode du jeu fictif qui tient compte de l'expérience antérieure d'après le poids qui diminue à mesure que cette expérience vieillit (cas de la mémoire amortie) est tout aussi inutile dans le cas général.

Les résultats obtenus par V. A. Volkonski [16] nous autorisent pourtant de formuler l'hypothèse suivante. La convergence du jeu fictif peut être apparemment assurée, et aussi, peut être accélérée, en remplaçant les formules (3.1), (3.2) de constitution récurrente des stratégies des adversaires par des formules du type

$$U^{(s+1)} = (1 - \lambda_{s+1})\, U^{(s)} + \lambda_{s+1} e_{i_{s+1}}, \tag{3.28}$$

$$W^{(t+1)} = (1 - \lambda_{t+1})\, W^{(t)} + \lambda_{t+1} e_{j_{t+1}}, \tag{3.29}$$

où

$$\lim_{s\to\infty} \lambda_s = 0; \quad \sum_{s=1}^{\infty} \lambda_s = \infty\ ; \ 0 \leqslant \lambda_s \leqslant 1. \tag{3.30}$$

Dans les formules (3.1), (3.2), $\lambda_s = \frac{1}{s}$; il est clair que dans ce cas les conditions (3.30) sont vérifiées. Si les coefficients de pondération décroissent comme $e^{-\alpha s}$, $\alpha > 0$, quand s croît, la deuxième des conditions (3.30) n'est pas observée et, en général, la procédure correspondante de résolution du jeu ne converge pas.

Il est peu probable qu'on puisse choisir d'avance une suite fixée λ_s, qui pour toute matrice de paiement garantira une accélération substantielle de la convergence, par rapport à la méthode de Brown. Il semble qu'on puisse accélérer la convergence en formant les coefficients de pondération λ_s au cours de la résolution du jeu. L'un des moyens possibles de formation des coefficients de pondération λ_s est proposé par V. Amvrossienko [3].

Les recommandations d'Amvrossienko sont fondées sur une expérience considérable de résolution des jeux par la méthode du jeu fictif. On a remarqué que dans la procédure itérative de la méthode du jeu fictif les adversaires appliquent un certain nombre de fois de suite les mêmes stratégies. Quand le numéro s de l'itération va en augmentant, le nombre de répétitions d'une même stratégie croît.

Lorsque le choix des stratégies par les deux adversaires est simultané, les répétitions commencent déjà à partir de s petits, et lorsqu'il est séquentiel, à partir de s relativement grands. On peut calculer le nombre de répétitions successives de mêmes coups par les joueurs et remplacer par une seule le nombre correspondant d'itérations. Alors, à chaque itération l'un des joueurs au moins appliquera une stratégie différente de celle de l'étape précédente. Le nombre total des itérations nécessaires pour atteindre la précision imposée de la solution du jeu diminue considérablement.

Lors du choix simultané des stratégies, le nombre l des répétitions successives (longueur de la série des stratégies qui se répètent) est déterminé après la s-ième itération par le plus petit des nombres l'_i et l''_j, où

$$l'_i = \begin{cases} 1 + \left[\dfrac{s\,(AW^{(s)})_{k''} - s\,(AW^{(s)})_i}{a_{ik'} - a_{k''k'}}\right] & \text{pour } a_{ik'} > a_{k''k'}, \\ \infty & \text{pour } a_{ik'} \leqslant a_{k''k'}, \end{cases}$$

$$l''_j = \begin{cases} 1 + \left[\dfrac{s\,(U^{(s)}A)_j - s\,(U^{(s)}A)_{k'}}{a_{k''k'} - a_{k''j}}\right] & \text{pour } a_{k''j} < a_{k''k'}, \\ \infty & \text{pour } a_{k''j} \geqslant a_{k''k'}\,; \end{cases}$$

$$i = 1,\ 2,\ \ldots,\ m\,;\ \ j = 1,\ 2,\ \ldots,\ n.$$

k'' est ici le numéro de l'élément maximal de la colonne $AW^{(s)}$; k', le numéro de l'élément minimal de la ligne $U^{(s)}A$; a_{ij}, les éléments de la matrice des gains initiale; $[M]$, la partie entière du nombre M.

Les l itérations peuvent être toutes remplacées par une seule. De plus, dans le schéma de la méthode du jeu fictif il convient d'ajouter à la ligne $sU^{(s)}A$ la ligne de la matrice des gains multipliée par l, caractérisant la stratégie reprise du premier joueur, et d'ajouter à la colonne $sAW^{(s)}$ la colonne de la matrice de paiement initiale multipliée par l, correspondant à la stratégie reprise du deuxième joueur. Compte tenu de la l-ième application des stratégies pures, il faut modifier de la manière correspondante les stratégies mixtes $U^{(s)}$ et $W^{(s)}$.

Le tableau 2.9 donne les 62 premières itérations de la résolution par la modification de la méthode du jeu fictif ici décrite, du jeu qui a été étudié précédemment par d'autres méthodes. Nous avons examiné ici le cas du choix simultané des stratégies par les deux joueurs. La comparaison du tableau 2.9 avec les tableaux 2.3 et 2.5 témoigne de l'efficacité de la modification proposée.

Tableau 2.9

		$sU^{(s)}A$			$sAW^{(s)}$			$U^{(s)}$			$W^{(s)}$			$v_1^{(s)}$	$v_2^{(s)}$	$\delta v^{(s)}$
N	s	1	2	3	1	2	3	1	2	3	1	2	3			
1	1	$\boxed{1}$	2	3	1	$\boxed{4}$	2	1	0	0	1	0	0	1	4	1,2
2	2	5	$\boxed{2}$	4	2	$\boxed{8}$	4	0,50	0,50	0	1	0	0	1	4	1,2
3	4	13	$\boxed{2}$	6	6	8	$\boxed{10}$	0,25	0,75	0	0,50	0,50	0	0,50	2,50	1,33
4	6	17	8	$\boxed{6}$	10	8	$\boxed{16}$	0,17	0,50	0,33	0,33	0,67	0	1	2,67	0,71
5	8	21	14	$\boxed{6}$	$\boxed{16}$	10	16	0,12	0,38	0,50	0,25	0,50	0,25	0,75	2	0,91
6	16	$\boxed{29}$	30	30	$\boxed{40}$	18	16	0,56	0,19	0,25	0,12	0,25	0,62	1,81	2,50	0,32
7	24	$\boxed{37}$	46	54	48	$\boxed{50}$	32	0,71	0,12	0,17	0,42	0,17	0,42	1,54	2,08	0,30
8	27	49	$\boxed{46}$	57	51	$\boxed{62}$	38	0,63	0,22	0,15	0,48	0,15	0,37	1,70	2,30	0,30
9	33	73	$\boxed{46}$	63	$\boxed{63}$	62	56	0,52	0,36	0,12	0,39	0,30	0,30	1,39	1,91	0,31
10	41	81	$\boxed{62}$	87	79	62	$\boxed{80}$	0,61	0,29	0,10	0,32	0,44	0,24	1,51	1,95	0,25
11	50	99	89	$\boxed{87}$	97	62	$\boxed{107}$	0,50	0,24	0,26	0,26	0,54	0,20	1,74	2,14	0,21
12	54	107	101	$\boxed{87}$	$\boxed{109}$	66	107	0,46	0,22	0,32	0,24	0,50	0,26	1,61	2,02	0,22
13	64	$\boxed{117}$	121	117	$\boxed{139}$	76	107	0,55	0,19	0,27	0,20	0,42	0,38	1,83	2,17	0,17
14	86	$\boxed{139}$	165	183	161	$\boxed{164}$	151	0,66	0,14	0,20	0,41	0,31	0,28	1,62	1,91	0,17
15	93	167	$\boxed{165}$	190	168	$\boxed{192}$	165	0,61	0,20	0,18	0,45	0,29	0,26	1,77	2,06	0,15
16	103	207	$\boxed{165}$	200	188	192	$\boxed{195}$	0,55	0,28	0,17	0,41	0,36	0,23	1,60	1,89	0,17

17	115	231	201	200	212	192	231	0,50	0,25	0,25	0,36	0,43	0,21	1,74	2,01	0,14
18	122	245	222	200	233	199	231	0,47	0,24	0,29	0,34	0,40	0,25	1,64	1,91	0,15
19	144	267	266	266	299	221	231	0,55	0,20	0,25	0,29	0,34	0,37	1,85	2,08	0,12
20	145	268	268	269	301	221	234	0,55	0,20	0,25	0,29	0,34	0,37	1,85	2,08	0,12
21	172	295	322	350	328	329	288	0,62	0,17	0,21	0,40	0,29	0,31	1,72	1,91	0,11
22	179	323	322	357	335	357	302	0,60	0,20	0,20	0,42	0,28	0,30	1,80	2,00	0,103
23	190	367	322	368	357	357	335	0,56	0,25	0,19	0,40	0,32	0,28	1,69	1,88	0,103
24	213	390	368	437	403	357	404	0,61	0,22	0,17	0,36	0,39	0,25	1,68	1,90	0,094
25	235	434	434	437	447	357	470	0,55	0,20	0,25	0,32	0,45	0,23	1,85	2,00	0,080
26	237	438	440	437	449	365	474	0,55	0,20	0,25	0,33	0,45	0,22	1,84	2,00	0,081
27	246	456	467	437	476	374	474	0,53	0,19	0,28	0,32	0,43	0,25	1,78	1,94	0,085
28	256	446	487	467	506	384	474	0,55	0,18	0,27	0,30	0,41	0,28	1,82	1,98	0,082
29	289	499	553	566	539	516	540	0,60	0,16	0,24	0,38	0,37	0,25	1,73	1,87	0,079
30	301	523	589	566	551	564	564	0,57	0,16	0,27	0,41	0,35	0,24	1,74	1,87	0,076
31	316	583	589	581	566	624	594	0,55	0,20	0,26	0,44	0,34	0,23	1,84	1,97	0,071
32	324	615	589	589	590	632	594	0,53	0,22	0,25	0,43	0,33	0,25	1,82	1,97	0,070
33	337	667	589	602	616	632	633	0,51	0,25	0,24	0,41	0,35	0,24	1,75	1,88	0,072
34	342	677	604	602	626	632	648	0,51	0,24	0,25	0,40	0,36	0,23	1,76	1,90	0,072

Suite

		$sU^{(s)}A$			$sAW^{(s)}$			$U^{(s)}$			$W^{(s)}$			$v_1^{(s)}$	$v_2^{(s)}$	$\delta v^{(s)}$
N	s	1	2	3	1	2	3	1	2	3	1	2	3			
35	350	693	628	$\boxed{602}$	$\boxed{650}$	640	648	0,49	0,24	0,27	0,39	0,35	0,25	1,72	1,86	0,079
36	376	719	$\boxed{680}$	680	$\boxed{728}$	666	648	0,53	0,22	0,25	0,37	0,33	0,30	1,81	1,93	0,068
37	415	$\boxed{758}$	758	797	$\boxed{806}$	666	765	0,57	0,20	0,23	0,33	0,39	0,27	1,83	1,95	0,061
38	457	$\boxed{800}$	842	923	848	834	$\boxed{849}$	0,61	0,18	0,21	0,39	0,36	0,25	1,75	1,86	0,059
39	465	$\boxed{816}$	866	923	856	$\boxed{866}$	865	0,60	0,18	0,22	0,40	0,35	0,25	1,76	1,86	0,059
40	478	868	$\boxed{866}$	936	869	$\boxed{918}$	891	0,59	0,20	0,21	0,42	0,34	0,24	1,81	1,92	0,058
41	488	908	$\boxed{866}$	946	889	918	$\boxed{921}$	0,57	0,22	0,21	0,41	0,35	0,23	1,78	1,89	0,061
42	515	962	947	$\boxed{946}$	943	918	$\boxed{1002}$	0,54	0,21	0,25	0,39	0,39	0,22	1,84	1,94	0,058
43	535	1002	1007	$\boxed{946}$	$\boxed{1003}$	938	1002	0,52	0,20	0,28	0,38	0,37	0,25	1,77	1,88	0,058
44	563	$\boxed{1030}$	1063	1030	$\boxed{1087}$	966	1002	0,55	0,19	0,26	0,36	0,35	0,29	1,83	1,92	0,054
45	604	$\boxed{1071}$	1145	1153	1128	$\boxed{1130}$	1084	0,58	0,17	0,25	0,40	0,33	0,27	1,78	1,87	0,053
46	623	1147	$\boxed{1145}$	1172	1147	$\boxed{1206}$	1122	0,56	0,20	0,24	0,42	0,32	0,26	1,84	1,94	0,052
47	652	1263	$\boxed{1145}$	1201	1205	1206	$\boxed{1209}$	0,53	0,24	0,23	0,40	0,35	0,25	1,76	1,85	0,054
48	671	1301	1202	$\boxed{1201}$	1243	1206	$\boxed{1266}$	0,52	0,23	0,25	0,39	0,37	0,24	1,79	1,89	0,053

49	679	1317	1226	$\boxed{1201}$	$\boxed{1267}$	1214	1266	0,51	0,23	0,26	0,38	0,37	0,25	1,77	1,87	0,053
50	704	1342	$\boxed{1276}$	1276	$\boxed{1342}$	1239	1266	0,53	0,22	0,25	0,37	0,35	0,28	1,81	1,91	0,050
51	770	$\boxed{1408}$	1408	1474	$\boxed{1474}$	1239	1464	0,57	0,20	0,23	0,34	0,41	0,25	1,83	1,92	0,046
52	781	$\boxed{1419}$	1430	1496	1485	1283	$\boxed{1486}$	0,58	0,20	0,22	0,35	0,40	0,25	1,82	1,90	0,046
53	820	1497	1547	$\boxed{1496}$	1524	1439	$\boxed{1564}$	0,55	0,19	0,26	0,38	0,38	0,24	1,82	1,91	0,044
54	834	1525	1589	$\boxed{1496}$	$\boxed{1566}$	1453	1564	0,54	0,18	0,27	0,37	0,38	0,25	1,80	1,87	0,046
55	849	$\boxed{1540}$	1619	1541	$\boxed{1611}$	1468	1564	0,55	0,18	0,27	0,37	0,37	0,26	1,82	1,90	0,045
56	897	$\boxed{1588}$	1715	1685	1659	$\boxed{1660}$	1660	0,57	0,17	0,26	0,40	0,35	0,25	1,77	1,85	0,044
57	929	1716	$\boxed{1715}$	1717	1691	$\boxed{1788}$	1724	0,55	0,20	0,25	0,42	0,34	0,24	1,85	1,93	0,042
58	951	1804	$\boxed{1715}$	1739	1735	1788	$\boxed{1790}$	0,54	0,22	0,24	0,41	0,35	0,24	1,80	1,88	0,043
59	960	1822	1742	$\boxed{1739}$	1753	1788	$\boxed{1817}$	0,53	0,22	0,25	0,41	0,36	0,23	1,81	1,89	0,044
60	988	1866	1808	$\boxed{1739}$	$\boxed{1819}$	1810	1817	0,52	0,21	0,26	0,40	0,35	0,25	1,76	1,84	0,045
61	1052	$\boxed{1930}$	1936	1931	$\boxed{2011}$	1874	1817	0,55	0,20	0,25	0,37	0,33	0,29	1,83	1,91	0,041
62	1098	$\boxed{1976}$	2028	2069	2057	$\boxed{2058}$	1909	0,57	0,19	0,24	0,40	0,31	0,28	1,80	1,87	0,041

La modification de V. Amvrossienko correspond à la méthode de Brown avec des coefficients de pondération de la forme

$$\lambda_s = \frac{l_s}{\sum_{i=1}^{s} l_i},$$

où l_s est le nombre d'itérations « agglutinées » de la méthode du jeu fictif à la s-ième étape de la méthode modifiée.

Amvrossienko a obtenu le critère empirique suivant de l'erreur relative $\delta v^{(s)}$ de la détermination de la valeur du jeu après s étapes du « jeu fictif » modifié, en fonction de l'ordre n d'une matrice de paiement carrée dont les éléments sont répartis uniformément sur le segment [0, 1] :

$$\frac{v_1^{(s)} + v_2^{(s)}}{2} \delta v^{(s)} \leqslant 0{,}003 \frac{n^{2,26}}{s}.$$

Il existe également des modifications qui accélèrent la convergence de la méthode itérative de von Neumann. En comparant cette dernière méthode avec les méthodes finies de programmation linéaire, G. von Neumann note que bien que les estimations *a priori* de la méthode du simplexe soient moins favorables que le critère de la convergence de la méthode itérative, l'expérience fait donner la préférence aux méthodes finies de programmation linéaire. On est toutefois fondé à s'attendre que la procédure itérative de von Neumann puisse être accélérée par divers procédés, qui se ramènent à un lissage de la suite d'itérations avec une certaine mémoire finie.

§ 4. Méthodes itératives de programmation linéaire

4.1. En plus des méthodes itératives de programmation linéaire obtenues par transformation des méthodes correspondantes de résolution des jeux, il existe des méthodes itératives établies directement pour le calcul des programmes optimaux des problèmes extrémaux linéaires conditionnels. De telles méthodes sont examinées dans [84], [85], [69], [10], [62] et dans d'autres travaux. Dans certaines de ces méthodes, par exemple dans celles d'Udzawa [69] et de Boulavski [10], la procédure itérative impose l'inclusion dans la fonction à optimiser d'une « faible » non-linéarité et la résolution de problèmes auxiliaires de programmation quadratique par des méthodes spécialement étudiées à cette fin. Nous omettons ici l'exposé de ces méthodes, qui nécessite de se référer aux fondements de la programmation convexe.

Dans la méthode de Thomas Pietszykowski [61], le problème de la programmation linéaire est ramené à un problème extrémal convexe inconditionnel équivalent qui peut être résolu par la méthode des gradients. Une certaine expérience accumulée par le Centre de

Calcul de l'Académie des sciences de l'U.R.S.S. dans la résolution des problèmes de programmation linéaire par la méthode de Pietszykowski atteste de l'efficacité de cette méthode. Ecrivons le problème de programmation linéaire sous la forme suivante, commode pour la méthode considérée

$$L = (C, X) \to \max; \tag{4.1}$$

$$\varphi_i(X) = (A^{(i)}, X) - b_i \geqslant 0, \ i = 1, 2, \ldots, m; \tag{4.2}$$

$$\varphi_{m+j}(X) = x_j = (e_j, X) \geqslant 0, \ j = 1, 2, \ldots, n_1 \leqslant n. \tag{4.3}$$

$A^{(i)}$ est ici le i-ème vecteur-ligne de la matrice des conditions; e_j, le vecteur unitaire de dimension n avec l'unité en j-ième position.

Posant

$$A^{(m+i)} = e_i, \ b_{m+i} = 0, \ i = 1, 2, \ldots, n_1,$$

récrivons les conditions (4.2)-(4.3) sous une forme équivalente

$$\varphi_i(X) = (A^{(i)}, X) - b_i \geqslant 0, \ i = 1, 2, \ldots, m + n_1. \tag{4.4}$$

Considérons dans ce paragraphe et les paragraphes suivants que $C \neq 0$ et que le domaine de définition de la forme linéaire est borné. Appelons *polyèdre* W l'ensemble convexe défini par les conditions (4.4).

Examinons la fonctionnelle

$$G_\mu(X) = (C, X) + \frac{\mu}{2} \sum_{i=1}^{m+n_1} \tilde{\delta}_i[\varphi_i(X)][\varphi_i(X)]^2, \tag{4.5}$$

où

$$\tilde{\delta}_i[\varphi_i(X)] = \delta_i(X) = \begin{cases} 0, \text{ si le point } X \text{ satisfait à la } i\text{-ème condition du problème;} \\ -1, \text{ si le point } X \text{ ne satisfait pas à la } i\text{-ème condition du problème.} \end{cases}$$

Pour les conditions du problème notées sous la forme (4.4)

$$\delta_i(X) = \begin{cases} 0 \text{ avec } \varphi_i(X) \geqslant 0, \\ -1 \text{ avec } \varphi_i(X) < 0. \end{cases}$$

On voit que la fonctionnelle $G_\mu(X)$ coïncide avec la forme linéaire (4.1) dans le domaine de sa définition (4.4). A l'extérieur du polyèdre W, la fonctionnelle $G_\mu(X)$ diffère de la forme linéaire (C, X) à la valeur de la forme quadratique $\frac{\mu}{2} \sum_{i \in I} [(A^{(i)}, X) - b_i]^2$, où I est l'ensemble des numéros des contraintes compromises. La courbe représentative de $G_\mu(X)$ dans le cas unidimensionnel est donnée pour différents μ par la figure 2.3. On adopte ici le segment $[a, b]$ comme domaine de définition de la forme linéaire.

Il est clair — et la courbe représentative le confirme — que, d'une façon générale, l'ensemble M des solutions du problème (4.1)-(4.3) ne coïncide pas avec l'ensemble M_μ des points en lesquels la fonctionnelle G_μ atteint un maximum dans tout l'espace E_n. Mais à mesure que μ croît, l'ensemble M_μ se rapproche de l'ensemble M.

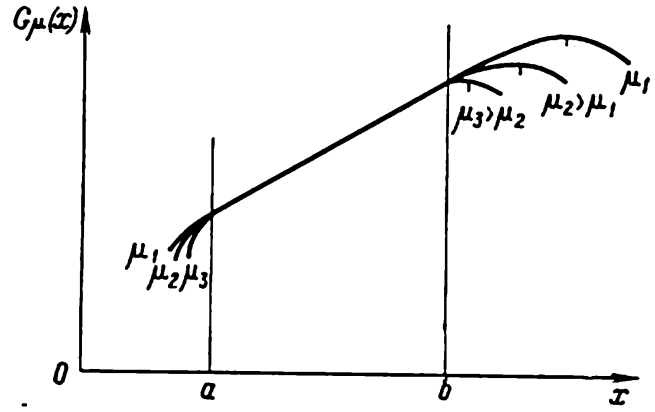

Fig. 2.3

Désignons par $\rho(X, M)$ la borne inférieure des distances entre X et $Y \in M$ pour tous les points Y de l'ensemble M. Dans [61] on trouve la démonstration de la proposition suivante.

Théorème 4.1. *Pour tout* $\varepsilon > 0$, *il existe un* $\mu_0 > 0$ *tel que pour tout* $\mu > \mu_0$ *l'inégalité*

$$\rho(X, M) < \varepsilon$$

soit vérifiée pour tout $X \in M_\mu$.

Il résulte du théorème 4.1 que la résolution du problème de programmation linéaire (dans les limites d'une précision donnée) se ramène au calcul de l'extrémum inconditionnel de la fonctionnelle $G_\mu(X)$. La solution la plus avantageuse est donnée par la méthode des gradients.

Définissons la suite $\{Z^{(k)}\}$ par la relation de récurrence suivante :

$$Z^{(k+1)} = Z^{(k)} + \theta^{(k)} g^{(k)}, \tag{4.6}$$

où $Z^{(0)}$ est un point quelconque de E_n ;

$g^{(k)}$, le gradient de la fonction $G_\mu X$ en $Z^{(k)}$.

On prend pour étape $\theta^{(k)}$ de la méthode des gradients une valeur θ qui maximise la fonction

$$\zeta(\theta) = G_\mu(Z^{(k)} + \theta g^{(k)}). \tag{4.7}$$

La fonction $\zeta(\theta)$ est convexe vers le haut. L'étape $\theta^{(k)}$ est donc la racine unique de l'équation

$$\zeta'(\theta) = 0.$$

En dérivant (4.7), on a

$$\zeta'(\theta)=\frac{\partial}{\partial\theta}G_\mu(Z^{(k)}+\theta g^{(k)})=(C,\ g^{(k)})+\mu\sum_{i=1}^{m+n_1}\delta_i(Z^{(k)}+$$
$$+\theta g^{(k)})[\varphi_i(Z^{(k)})+\theta(A^{(i)},\ g^{(k)})](A^{(i)},\ g^{(k)}).$$

On voit sans peine que $\zeta'(\theta)$ est une fonction linéaire par morceaux de la variable θ.

Pour calculer la racine de l'équation $\zeta'(\theta)=0$, il est rationnel de disposer par ordre de croissance les points

$$\vartheta_i^{(k)}=-\frac{\varphi_i(Z^{(k)})}{(A^{(i)},\ g^{(k)})},$$

où la fonction $\zeta'(\theta)$ subit une brisure. Le point $\theta^{(k)}$ appartient au segment $[\vartheta_l^{(k)},\ \vartheta_{l+1}^{(k)}]$ pour lequel

$$\zeta'(\vartheta_l^{(k)})>0 \text{ et } \zeta'(\vartheta_{l+1}^{(k)})\leqslant 0;$$

$$\theta^{(k)}=\frac{\zeta'(\vartheta_l^{(k)})\vartheta_{l+1}^{(k)}-\zeta'(\vartheta_{l+1}^{(k)})\vartheta_l^{(k)}}{\zeta'(\vartheta_l^{(k)})-\zeta'(\vartheta_{l+1}^{(k)})}.$$

Ce choix de l'étape assure la convergence de la procédure itérative.

Partout où il faut insister sur la correspondance de la suite $\{Z^{(k)}\}$ à une valeur fixée du paramètre μ, nous la désignerons par $\{Z_\mu^{(k)}\}$.

On a la proposition suivante.

Théorème 4.2. *La suite $\{Z_\mu^{(k)}\}$ définie par la relation (4.6) jouit, quels que soient $Z^{(0)}\in E_n$ et $\mu>0$, de la propriété suivante:*

$$\lim_{k\to\infty}\rho(Z_\mu^{(k)},\ M_\mu)=0.$$

D'après le théorème 4.2, avec $\mu>0$ fixés et k grands, les éléments de la suite $\{Z_\mu^{(k)}\}$ s'approchent suffisamment près des points de l'ensemble M_μ.

Selon le théorème 4.1, pour des valeurs de μ suffisamment grandes, tout point de M_μ est proche de l'ensemble des solutions du problème de programmation linéaire. On en déduit qu'avec μ et k suffisamment grands, on peut s'approcher aussi près que l'on veut de la solution du problème (4.1)-(4.3).

Décrivons maintenant le schéma de calcul de la procédure de construction de la suite itérative $\{Z^{(k)}\}$.

1. Calculer

$$\beta_i^{(k)}=\varphi_i(Z^{(k)}),\ i=1,\ 2,\ \ldots,\ m+n_1.$$

2. Déterminer les composantes du gradient $g^{(k)}$:

$$g^{(k)} = C + \mu \sum_{i=1}^{m+n_1} \delta_i (Z^{(k)}) \beta_i^{(k)} A^{(i)}.$$

3. Si toutes les composantes $g_j^{(k)}$ du gradient $g^{(k)}$ ne dépassent pas en valeur absolue le petit nombre imposé $\varepsilon > 0$, $Z^{(k+1)} \approx Z^{(k)}$ et la procédure de résolution du problème pour la valeur adoptée du paramètre est terminée. Avec $| g_j^{(k)} | > \varepsilon$,

$$\xi_i^{(k)} = (A^{(i)}, g^{(k)}).$$

4. Calculer les rapports

$$\vartheta_i^{(k)} = -\frac{\beta_i^{(k)}}{\xi_i^{(k)}}, \quad i = 1, 2, \ldots, m+n_1, \quad \xi_i^{(k)} \neq 0,$$

et les ranger par ordre de croissance:

$$\vartheta_{i_1}^{(k)} \leqslant \vartheta_{i_2}^{(k)} \leqslant \ldots \leqslant \vartheta_{i_s}^{(k)}.$$

5. Déterminer les valeurs de la fonction

$$\zeta'(\theta) = \frac{\partial}{\partial\theta} G_\mu (Z^{(k)} + \theta g^{(k)}) = (C, g^{(k)}) +$$

$$+ \mu \sum_{i=1}^{m+n_1} \delta_i (Z^{(k)} + \theta g^{(k)}) (\beta_i^{(k)} + \theta \xi_i^{(k)}) \xi_i^{(k)}$$

en $\vartheta_{i_l}^{(k)}$.

6. Calculer l'étape $\theta^{(k)}$

$$\theta^{(k)} = \frac{\zeta'(\vartheta_{i_l}^{(k)}) \vartheta_{i_{l+1}}^{(k)} - \zeta'(\vartheta_{i_{l+1}}^{(k)}) \vartheta_{i_l}^{(k)}}{\zeta'(\vartheta_{i_l}^{(k)}) - \zeta'(\vartheta_{i_{l+1}}^{(k)})},$$

où $\vartheta_{i_l}^{(k)}$ et $\vartheta_{i_{l+1}}^{(k)}$ déterminent le segment sur lequel $\zeta'(\theta)$ change de signe, c'est-à-dire $\zeta'(\vartheta_{i_l}^{(k)}) > 0$ alors que $\zeta'(\vartheta_{i_{l+1}}^{(k)}) \leqslant 0$.

7. Construire

$$Z^{(k+1)} = Z^{(k)} + \theta^{(k)} g^{(k)}.$$

Il est facile de vérifier que les valeurs de $\beta_i^{(k)}$ peuvent être calculées d'après les formules de récurrence suivantes:

$$\beta_i^{(k+1)} = \beta_i^{(k)} + \theta^{(k)} \xi_i^{(k)}.$$

Pour résoudre un problème de programmation linéaire par la méthode considérée, le mieux est de commencer les calculs à partir d'une valeur de μ relativement petite. Pour des μ grands, la quantité $\rho(Z_\mu^{(k)}) M_\mu$ tend lentement vers zéro. Si, en vertu des condi-

tions $|g_j^{(k)}| \leqslant \varepsilon$, la procédure itérative s'interrompt au vecteur $Z_\mu^{(k_0)}$ associé aux variables d'écart négatives $\beta_i^{(k_0)} = \varphi(Z_\mu^{(k_0)})$ des contraintes du problème, variables grandes en valeur absolue, il faut augmenter alors μ et reprendre la procédure de construction de l'extrémum de la fonctionnelle $G_\mu(X)$. La procédure s'achève avec les μ et k tels qu'ils vérifient simultanément les conditions

$$|g_j^{(k)}| \leqslant \varepsilon, \quad j = 1, 2, \ldots, n;$$

$$-\varphi_i(Z_\mu^{(k)}) \leqslant \varepsilon_1, \quad i = 1, 2, \ldots, m + n_1,$$

où $\varepsilon > 0$ et $\varepsilon_1 < 0$ sont les petits nombres donnés qui définissent les approximations admissibles du programme optimal du problème de programmation linéaire.

Les tableaux 2.10-2.12 fournissent les principales étapes de la procédure de la méthode de Pietszykowski, appliquée à l'exemple étudié dans ce qui précède par d'autres méthodes (cf., par exemple, § 3.1). Pour conserver les notations adoptées dans le présent paragraphe, les conditions du problème sont mises sous la forme (4.4). Les tableaux contiennent les composantes du vecteur $Z^{(k)}$, les valeurs des variables d'écart $\beta_i^{(k)}$ des conditions du problème, les composantes $g_j^{(k)}$ du vecteur du gradient $g^{(k)}$ de la fonctionnelle $G_\mu(Z^{(k)})$ et la valeur de l'étape $\theta^{(k)}$ de la méthode des gradients pour $\mu = 1$, 2 et 10. L'indice k est le numéro de l'itération.

La procédure part du point $Z^{(0)} = (0, 0, 0)$. Dans toutes les itérations réalisées, les composantes $Z^{(k)}$ se sont avérées non négatives. C'est pourquoi les variables d'écart dans les conditions $x_j \geqslant 0$ ne sont pas indiquées dans les tableaux.

Adoptons pour caractéristiques de l'approximation admissible les valeurs $\varepsilon = 0{,}007$ et $\varepsilon_1 = 0{,}03$. D'après les tableaux, avec $\mu = 1$ et $\mu = 2$, lorsque les variables d'écart des conditions du problème sont assez grandes en valeur absolue, les composantes g_j deviennent plus petites que ε (comparées aux composantes du vecteur des contraintes). Lorsque $\mu = 10$ après la cinquième itération, les composantes $g_j^{(k)}$ du vecteur du gradient sont des grandeurs de l'ordre de 0,005 à 0,007 et les variables d'écart des conditions n'excèdent pas 0,03. Le vecteur

$$Z^{(5)} = (0{,}217;\ 0{,}192;\ 0{,}143).$$

La solution exacte du problème (cf. § 3.1) est égale à

$$X^* = \left(\frac{8}{37}, \frac{7}{37}, \frac{5}{37}\right) \approx (0{,}216;\ 0{,}189;\ 0{,}135).$$

La valeur optimale de la forme linéaire

$$L(X^*) = \frac{20}{37} \approx 0{,}541.$$

La valeur de la fonctionnelle $G_\mu(X)$ sur le vecteur $Z^{(5)}$

$$G_\mu(Z^{(5)}) = G_{10}(Z^{(5)}) \approx 0{,}546.$$

Les variables d'écart des conditions du problème pour le vecteur $Z^{(5)}$ sont

$$\{-0{,}03;\ -0{,}01;\ -0{,}01;\ 0;\ 0;\ 0\}.$$

4.2. Dans la méthode de Pietszykowski, la solution d'un problème extrémal conditionnel se ramène à la solution d'un problème inconditionnel. La sortie d'un point hors du domaine admissible entraîne une pénalité qui diminue la valeur de la forme linéaire maximisée.

Cette pénalité déterminée par les variables d'écart permet d'utiliser les méthodes extrémales inconditionnelles pour calculer le maximum d'une fonctionnelle dans un certain domaine fixé. En changeant la relation entre la pénalité et la valeur de la variable d'écart, on peut obtenir différentes méthodes de résolution des problèmes de programmation mathématique. Dans la méthode de Pietszykowski, la relation entre la pénalité Φ et la valeur de la variable d'écart φ est donnée par la fonction

$$\Phi(\varphi) = \sum_{i=1}^{m+n_1} \Phi_i(\varphi_i) = \frac{\mu}{2} \sum_{i=1}^{m+n_1} \tilde{\delta}_i(\varphi_i)\,[\varphi_i(X)]^2. \tag{4.8}$$

La justification de la convergence de la méthode ne subira pas de grands changements par rapport à la démonstration de la convergence de la méthode de Pietszykowski si l'on remplace les fonctions de pénalité Φ_i du type (4.8) par n'importe quelles autres fonctions lisses décroissantes monotonement lorsque la variable d'écart φ_i croît et égales à zéro lorsque les conditions du problème sont respectées.

Au § 4.1 nous avons examiné une variante discrète de la méthode de Pietszykowski, conçue pour être réalisée sur ordinateur. Il est aisé de construire un analogue continu de cette méthode, adapté à un calculateur digital.

A mesure que t augmente, le point $X(t)$ qui vérifie l'équation différentielle vectorielle

$$\frac{dX}{dt} = \operatorname{grad} G_\mu(X), \tag{4.9}$$

s'approche du point du maximum de $G_\mu(X)$. Avec μ suffisamment grand, le point du maximum de $G_\mu(X)$ devient aussi proche que l'on veut de la solution du problème correspondant de programmation linéaire.

Pour la méthode de Pietszykowski,

$$\frac{dX}{dt} = C + \mu \sum_{i=1}^{m+n_1} \tilde{\delta}_i(\varphi_i)\,\varphi_i(X)\,A^{(i)}, \tag{4.10}$$

Tableau 2.10

$\mu = 1$

i, j / n° de l'itération	z_j			β_i			g_j			θ_k
	1	2	3	1	2	3	1	2	3	—
0	0	0	0	1	1	1	1	1	1	0,2209
1	0,2209	0,2209	0,2209	−0,3256	−0,1046	−0,1046	0,0465	0,0349	−0,0814	0,1509
2	0,2279	0,2262	0,2087	−0,3064	−0,1204	−0,1343	−0,0566	−0,0158	−0,0395	0,0360
3	0,2259	0,2256	0,2073	−0,2989	−0,1108	−0,1286	0,0008	0,0165	−0,0076	0,1204
4	0,2260	0,2276	0,2064	−0,3002	−0,1102	−0,1347	−0,0106	−0,0045	−0,0110	0,0368
5	0,2256	0,2274	0,2060	−0,2983	−0,1083	−0,1334	0,0018	0,0032	−0,0032	0,1
6	0,2258	0,2277	0,2056	−0,2982	−0,1087	−0,1347	−0,0024	−0,0005	−0,0032	0,04
7	0,2257	0,2277	0,2055							

$\mu = 2$

n° de l'itération \ i, j	z_j			β_i			g_j			θ_k
	1	2	3	1	2	3	1	2	3	—
0	0	0	0	1	1	1	0,5	0,5	0,5	0,4067
1	0,2033	0,2033	0,2033	—0,2200	—0,0167	—0,0167	—0,18	0,01	—0,1767	0,1145
2	0,2239	0,2044	0,1831	—0,1820	—0,0787	—0,0610	—0,119	—0,047	—0,125	0,0376
3	0,2199	0,2027	0,1777	—0,1583	—0,0572	—0,0478	0,0173	0,0402	—0,0320	0,1153
4	0,2219	0,2073	0,1740	—0,1584	—0,0615	—0,0657	—0,0358	—0,0139	—0,0369	0,0374
5	0,2205	0,2068	0,1726	—0,1519	—0,0548	—0,0614	0,0062	0,0119	—0,0107	0,1160
6	0,2212	0,2082	0,1714	—0,1517	—0,0564	—0,0670	—0,0111	—0,0043	—0,0114	0,0370
7	0,2208	0,2080	0,1710	—0,1497	—0,0543	—0,0657	0,0017	0,0036	—0,0034	0,15
8	0,2211	0,2085	0,1704	—0,1494	—0,0548	—0,0678	—0,0044	—0,0024	—0,0032	0,03
9	0,2210	0,2085	0,1703							

Tableau 2.12

$\mu = 10$

n° de l'itération \ i, j	z_j			β_i			g_i			θ_k
	1	2	3	1	2	3	1	2	3	—
0	0	0	0	1	1	1	0,1	0,1	0,1	1,75
1	0,175	0,175	0,175	—0,05	0,125	0,125	0,05	0	—0,05	0,8333
2	0,2167	0,175	0,1333	0,0333	0	0,0417	0,1	0,1	0,1	0,0820
3	0,2249	0,1832	0,1415	—0,0158	—0,0410	0,0007	—0,0798	0,0683	0,0115	0,1066
4	0,2164	0,1905	0,1428	—0,0256	—0,0082	—0,0042	0,0333	0,0364	0,0151	0,0351
5	0,2175	0,1918	0,1433	—0,0309	—0,0134	—0,0103	—0,0052	0,0072	0,0061	0,1257
6	0,2169	0,1927	0,1425	—0,0298	—0,0100	—0,0118	0,0065	0,0052	0,0006	0,0370
7	0,2171	0,1929	0,1425	—0,0305	—0,0110	—0,0128	—0,0002	0,0006	—0,0024	0,1208
8	0,2171	0,1929	0,1422	—0,0297	—0,0106	—0,0130	0,0019	0,0017	0,0003	0,0368
9	0,2172	0,1930	0,1423	—0,0299	—0,0109	—0,0133	—0,0001	0,0003	—0,0007	0,1250

où comme auparavant le vecteur $A^{(i)} = (a_{i1}, \ldots, a_{in})$ est normal à l'hyperplan défini par la i-ème condition du système (4.4) et orienté vers l'intérieur du domaine de définition du problème.

En changeant les fonctions de pénalité on peut obtenir à partir du schéma (4.9) différentes méthodes itératives de résolution du problème de programmation linéaire. La démonstration de la convergence de ces méthodes avec les contraintes mentionnées sur le choix de la fonction de pénalité ne diffère pas, en principe, de la justification de la convergence des réalisations discrètes de la méthode de Pietszykowski.

Les raisonnements exposés peuvent tous être étendus également au cas de la programmation convexe.

I. B. Pyne [60] a proposé une méthode de programmation linéaire qui s'inscrit dans le cadre du schéma (4.9). Cependant, la fonction de pénalité choisie par Pyne n'est pas lisse. Dans cette méthode,

$$\Phi_i(\varphi_i) = \tilde{\delta}_i(\varphi_i)\,\varphi_i(X). \tag{4.11}$$

Pour une fonction de pénalité du type (4.11), l'équation différentielle (4.9) s'écrit

$$\frac{dX}{dt} = C - \mu \sum_{i=1}^{m+n_1} \tilde{\delta}_i(\varphi_i)\,A^{(i)}. \tag{4.12}$$

Les composantes du vecteur $X(t)$ satisfont au système d'équations différentielles

$$\frac{dx_j}{dt} = c_j - \mu \sum_{i=1}^{m+n_1} a_{ij}\delta_i(X). \tag{4.13}$$

Les dérivées de la fonction de pénalité dans la méthode de Pyne sont des fonctions discontinues. Aussi la convergence de la méthode de Pyne ne découle pas des raisonnements correspondants appliqués aux analogues continus de la méthode de Pietszykowski aux fonctions de pénalité lisses. Les conditions de convergence de la méthode de Pyne doivent être établies de façon indépendante.

Pour résoudre le système d'équations différentielles (4.9), on peut utiliser des installations de simulation analogique du type MN-7 ou MN-8. Pourtant pour l'analyse de problèmes d'une taille importante, un modèle composé de blocs de MN-7 ou de MN-8 peut s'avérer trop complexe et nécessiter des puissances élevées. Il semble que la mise au point des amplificateurs à semi-conducteurs, susceptibles d'assurer la résolution du système (4.9), ait le plus d'avenir.

La possibilité d'automatiser la résolution des problèmes de programmation linéaire (et pas seulement linéaire) peut être nettement étendue par l'utilisation de schémas impliquant des calculateurs analogiques et digitaux. Certaines recommandations dans ce sens sont données dans [18].

4.3. L'analyse numérique montre que dans les méthodes itératives de programmation linéaire (surtout dans celles obtenues par transformation des méthodes correspondantes de résolution des jeux), la vitesse à laquelle la forme linéaire s'approche de l'optimum décroît nettement à mesure que le nombre d'itérations augmente. Autrement dit, l'expérience des calculs montre que les méthodes itératives permettent en un nombre d'étapes relativement petit d'obtenir une certaine approximation de la valeur optimale de la forme linéaire, qui ensuite ne devient que très lentement plus précise.

On en tire deux conclusions qualitatives.

1. L'utilisation de méthodes itératives de résolution des jeux pour calculer un programme optimal n'est apparemment rationnelle que dans le cas où l'on a besoin d'obtenir assez vite une approximation grossière de la solution du problème.

2. Il se peut que l'application des méthodes combinées qui mettent en valeur les mérites des méthodes itératives et finies, soit avantageuse pour la résolution des jeux et des problèmes de programmation linéaire de grande taille. La solution approchée d'un jeu, fournie par une procédure itérative en un nombre d'étapes relativement petit, permet de déterminer les programmes du couple correspondant de problèmes duals de programmation linéaire.

La méthode des gradients, exposée dans [87] (chapitre 9, § 5), permet de passer de chacun de ces programmes à un programme de base du problème correspondant de programmation linéaire.

Il faut s'attendre à ce que le choix du programme de base ainsi calculé comme programme de base initial permette de diminuer le nombre d'étapes nécessaires pour résoudre le problème à l'aide d'une des méthodes finies de programmation linéaire.

Exercices

1. Montrer que tous les points-selles d'une matrice de paiement déterminent la même valeur du jeu.

2. Démontrer l'inégalité

$$\max \min \| a_{ij} \| \leqslant \min \max \| a_{ij} \|.$$

Montrer que cette inégalité se transforme en égalité si et seulement si la matrice de paiement possède un point-selle.

3. Démontrer l'inégalité

$$\max \min M \leqslant \min \max M,$$

où $M = M(U, W)$ est la fonction de paiement du jeu. Montrer que cette inégalité se transforme en égalité si et seulement si le jeu possède un point-selle (U^*, W^*).

4. Soit $\| a_{ij} \| \geqslant 0$ la matrice de paiement d'un jeu. Montrer que pour que la valeur du jeu v soit positive il faut et il suffit que chaque colonne de la matrice compte au moins un élément positif.

5. Considérons le problème de programmation linéaire

$$\sum_{j=1}^{n} c_j x_j \to \max,$$

$$\sum_{j=1}^{m} a_{ij} x_j \leqslant b_i, \quad i=1, 2, \ldots, m;$$

$$x_j \geqslant 0, \quad j=1, 2, \ldots, n.$$

Supposons que $a_{ij} \geqslant 0$, $b_i > 0$, $c_j > 0$, $i = 1, \ldots, m$; $j = 1, \ldots, n$, et que chaque colonne de la matrice $\| a_{ij} \|$ compte au moins un élément positif. Montrer que si $U^* = (u_1^*, \ldots, u_m^*)$ et $W^* = (w_1^*, \ldots, w_n^*)$ sont des stratégies optimales des joueurs et v la valeur du jeu à matrice des gains

$$\| \tilde{a}_{ij} \| = \left\| \frac{a_{ij}}{b_i c_j} \right\|,$$

les vecteurs X^* et Y^* aux composantes

$$x_j^* = \frac{w_j^*}{c_j v}, \qquad y_i^* = \frac{u_i^*}{b_i v}, \qquad i=1, 2, \ldots, m; \qquad j=1, 2, \ldots, n,$$

sont solutions du problème considéré de programmation linéaire et de son dual.

6. Construire un jeu matriciel équivalent au problème de programmation linéaire

$$\begin{aligned}
3x_1 + x_2 + 2x_3 + 5x_4 + 3x_5 &\to \max,\\
2x_1 + x_2 + 3x_4 + 2x_5 &\leqslant 4,\\
3x_2 + x_3 + 2x_4 + 5x_5 &\leqslant 9,\\
4x_1 + 2x_2 + 5x_4 + 3x_5 &\leqslant 8,\\
x_j \geqslant 0,\ j &= 1, 2, 3, 4, 5.
\end{aligned}$$

7. Soit un jeu matriciel à matrice de paiement

$$\begin{Vmatrix} 2 & 3 & 5 \\ 6 & 4 & 2 \\ 15 & 8 & 0 \end{Vmatrix}.$$

Construire un couple équivalent de problèmes duals de programmation linéaire.

8. Montrer que la condition nécessaire et suffisante pour qu'une stratégie pure e_k du premier joueur soit essentielle est que l'égalité

$$e_k A W^* = v$$

soit vérifiée par toutes les stratégies optimales W^* du deuxième joueur.

9. Montrer que dans tout jeu matriciel, la différence entre les nombres $\bar{m}$, $\bar{n}$ de stratégies essentielles des joueurs est égale à la différence entre les dimensions ρ_u, ρ_w des polyèdres des stratégies optimales des deux adversaires.

(Indication : il résulte du théorème 1.2 du chapitre 5 de [87] et de l'exercice 8 que $\rho_u = \bar{m} - \rho$, $\rho_w = \bar{n} - \rho$, où ρ est le rang de la matrice composée des éléments qui figurent à l'intersection des lignes essentielles et des colonnes essentielles de la matrice du jeu.)

10. Démontrer la proposition 4° du § 2.3 du présent chapitre.

11. Composer le schéma bloc de la résolution d'un jeu à l'aide du deuxième algorithme de la méthode du simplexe.

12. Composer le schéma de résolution d'un jeu à l'aide d'une méthode combinée utilisant la méthode du jeu fictif et la méthode des gradients pour calculer le programme de base initial, et le deuxième algorithme du simplexe pour déterminer le programme optimal du problème correspondant de programmation linéaire (cf. § 4.3).

13. Composer un schéma de calcul pour la résolution du problème de programmation linéaire par la méthode combinée de l'exercice 12.

14. Résoudre le jeu

$$\begin{Vmatrix} 3 & 2 & 3 & 1 & 6 \\ 4 & 4 & 0 & 4 & 3 \\ 1 & 5 & 3 & 6 & 4 \\ 3 & 2 & 4 & 3 & 2 \end{Vmatrix}$$

l'aide du deuxième algorithme de la méthode du simplexe.

15. Comparer le nombre d'itérations nécessaires pour obtenir une solution approchée du jeu de l'exercice 14 (jusqu'à l'obtention de $\delta v = 0{,}05$) avec les variantes suivantes de la méthode du jeu fictif:

a) choix simultané des stratégies;

b) choix séquentiel des stratégies;

c) choix simultané des stratégies et application de la modification (cf. § 4) assurant à chaque itération le changement de stratégie par au moins l'un des adversaires.

16. Trouver une solution approchée du jeu de l'exercice 14 en se servant de la méthode itérative de von Neumann. Se limiter au nombre d'itérations qui fournissent l'estimation $\Phi^{(s)} < 0{,}1$.

17. Résoudre à l'aide d'une méthode combinée (cf. exercices 12 et 13) le problème suivant de programmation linéaire:

$$\begin{aligned} x_1 + 2x_2 + 3x_3 + 4x_4 + 5x_5 &\to \max, \\ 2x_1 \qquad\quad + x_3 + 3x_4 + x_5 &\leqslant 3, \\ x_1 + 3x_2 + 2x_3 + x_4 \qquad\quad &\leqslant 6, \\ x_1 + 2x_2 + 4x_3 + 2x_4 + 3x_5 &\leqslant 7, \\ x_j \geqslant 0,\ j = 1,\ 2,\ 3,\ 4,\ 5. \end{aligned}$$

18. Montrer que le couple de problèmes duals de programmation linéaire (2.12)-(2.13) est équivalent à un jeu matriciel à matrice de paiement A.

Chapitre 3

PROGRAMMATION PARAMÉTRIQUE

La programmation paramétrique est une branche particulière de la programmation mathématique, qui étudie la théorie et les méthodes économiques de résolution des problèmes extrémaux conditionnels dans lesquels les critères de qualité ou les conditions dépendent d'un ou de plusieurs paramètres.

On considère ici le cas de la programmation linéaire paramétrique.

Les §§ 1, 2 étudient en détail les problèmes dont les coefficients des formes linéaires ou les composantes des vecteurs des contraintes dépendent linéairement d'un seul paramètre (cas $C = C' + \lambda C''$ ou $B = B' + \mu B''$).

Le § 3 est consacré à l'examen de cas plus complexes. On y donne un algorithme de résolution des problèmes où les coefficients c_j et b_i dépendent linéairement d'un paramètre (généralisation des schémas de calcul des deux paragraphes précédents). On décrit ensuite une méthode de résolution du problème paramétrique dont un vecteur des conditions dépend linéairement d'un paramètre. Cette méthode se ramène aisément au cas où le paramètre entre linéairement dans l'une des lignes de la matrice des conditions du problème. A titre de conclusion du paragraphe, on expose le schéma général d'un algorithme de programmation linéaire paramétrique dans lequel le paramètre se trouve dans tous les éléments de la matrice complète des conditions du problème, la dépendance du paramètre n'étant pas seulement supposée linéaire (elle peut être définie par des fonctions analytiques quelconques).

Les premiers travaux de programmation paramétrique sont dus à S. I. Gass et T. Saaty [19], [20]. Ils se rapportent aux problèmes étudiés aux §§ 1 et 2. Signalons également l'ouvrage [44], consacré au problème de programmation linéaire dont l'un des vecteurs des conditions dépend linéairement d'un paramètre.

La nécessité de la programmation paramétrique est définie par de nombreuses circonstances. Les problèmes pratiques comptent en général plusieurs paramètres initiaux dont on connaît non pas les

valeurs précises mais certaines marges possibles de leur variation. C'est pourquoi l'étude de la façon dont le programme optimal d'un problème dépend des variations de certains paramètres des conditions de ce problème se trouve être souvent l'unique possibilité de motiver le choix de telle ou telle solution.

Ensuite, nombre de problèmes pratiques de planification et de gestion sont des problèmes dynamiques. Le mode le plus naturel de mise en évidence de la dynamique de la solution, qui consiste à calculer les programmes optimaux de plusieurs problèmes associés à des époques différentes, exige en général un travail excessif. Dans ces conditions, l'application des méthodes de la programmation paramétrique peut être très efficace.

Enfin, dans plusieurs cas, les organismes de planification et de gestion ou les réalisateurs de projets de systèmes jouissent d'une certaine liberté dans le choix des données de départ déterminant le problème de programmation mathématique auquel se ramènent la planification, la gestion ou l'élaboration du projet. Dans de tels cas, la programmation paramétrique constitue un outil utile, qui permet de calculer par une voie économique les moyens d'action admissibles les plus rationnels sur les conditions du problème. D'autres applications encore de la programmation paramétrique sont formulées au § 4. On y trouve notamment le procédé économique de résolution des problèmes de programmation linéaire qui diffèrent par une condition supplémentaire des modèles pour lesquels on dispose de méthodes d'analyse simples. Dans ce même paragraphe est exposé un algorithme pour la résolution des problèmes de programmation homographique.

La programmation paramétrique permet également d'estimer la stabilité de la solution des problèmes de programmation mathématique par rapport aux erreurs aléatoires dans les données initiales et aux défaillances des machines. L'estimation de la stabilité est liée directement à la question de la sensibilité de la solution des problèmes de programmation linéaire à une éventuelle variation des conditions du problème. Cette question fait l'objet du § 5.

§ 1. Cas $C = C' + \lambda C''$

1.1. Supposons que le polygone représenté figure 3.1 réponde au domaine de définition du problème de programmation linéaire et que la droite MN à vecteur directeur C corresponde à la forme linéaire du problème. En déplaçant parallèlement à elle-même la droite MN dans le sens d'un accroissement de la forme linéaire, on aboutit au sommet A, image géométrique du programme optimal. La figure montre que le sommet A du polygone des conditions ne fait pas que répondre à la solution du problème pour la forme linéaire définie par la droite MN à vecteur directeur C. Ce même sommet reste encore

optimal si l'on remplace MN par toute droite disposée entre les droites M_2N_2 et M_3N_3 ($M_2N_2 \parallel AB$, $M_3N_3 \parallel GA$).

Imaginons maintenant que les coefficients de la forme linéaire dépendent d'un certain paramètre λ, la droite MN correspondant à la

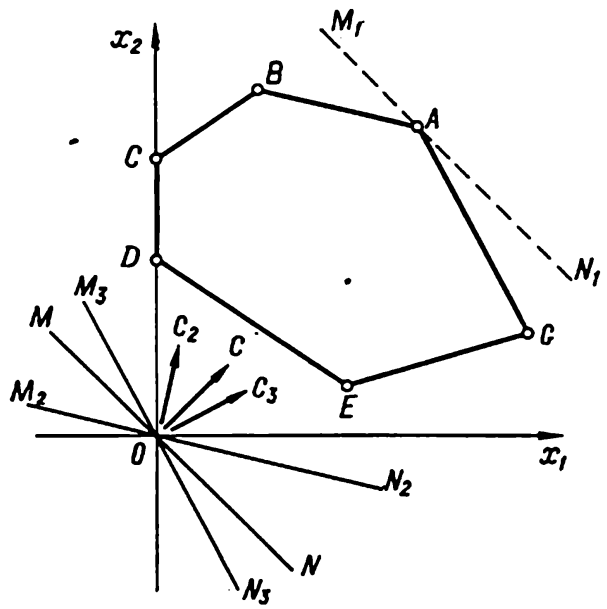

Fig. 3.1

valeur $\lambda = \lambda_0$, les droites M_2N_2 et M_3N_3 aux valeurs $\lambda = \underline{\lambda}$ et $\lambda = \overline{\lambda}$, et la variation de λ de $\underline{\lambda}$ à $\overline{\lambda}$ étant associée à une rotation de MN dans le sens des aiguilles d'une montre.

Lorsque $\lambda = \lambda_0$, le programme optimal du problème est associé au sommet A, et la valeur optimale de la forme linéaire à la droite M_1N_1. Modifions progressivement la valeur du paramètre λ. La droite MN pivote alors autour du point O. Toutefois, tant que son vecteur directeur C ne sortira pas de l'angle formé par les vecteurs directeurs C_2 et C_3 des droites M_2N_2 et M_3N_3, le sommet A correspondra invariablement au programme optimal du problème. Avec $\lambda = \underline{\lambda}$ ($\lambda = \overline{\lambda}$), la droite MN devient parallèle au côté AB (GA) du polygone des conditions. Le problème correspondant de programmation linéaire a une solution qui n'est pas unique. Le côté AB (ou GA) du polygone est l'image de l'ensemble des programmes optimaux du problème considéré. La variation aussi petite que l'on veut de λ déterminera un problème de programmation linéaire à programme optimal nouveau. L'image de la nouvelle solution est le sommet B (ou G). La variation ultérieure de λ dans la marge définie laisse invariable le programme optimal. En poursuivant le processus ainsi décrit, on peut découper la marge imposée de variation de λ en un nombre fini

de parties, dont chacune correspond à une famille de formes linéaires qui atteignent toutes leur valeur maximale au même point du domaine de définition du problème.

Si, au lieu d'un polygone convexe, on considère un ensemble convexe, il est alors facile de définir à partir de considérations géométriques les limites de la marge de variation de λ qui correspond à une forme linéaire infinie dans son domaine de définition. Il en est de même dans les espaces de plus grande dimension.

Supposons qu'un sommet A de l'ensemble convexe des conditions corresponde à la forme linéaire optimale dont l'image géométrique est donnée par l'hyperplan H au vecteur directeur $C'+\lambda C''$. Lorsque λ varie, le vecteur directeur de l'hyperplan de la forme linéaire, qui contient le point A, subit une rotation, et pour une certaine valeur $\bar{\lambda}$ du paramètre λ l'hyperplan H peut balayer l'arête de l'ensemble (par exemple, l'arête AB). Avec $\lambda = \bar{\lambda}$, l'optimum de la forme linéaire s'obtient en chaque point de l'arête AB. Lors de la variation ultérieure de λ, on peut déterminer un intervalle dont tous les points correspondent à la même solution, qui a pour image géométrique le point B voisin de A. On observe toujours une telle situation dans le cas bidimensionnel. La procédure de résolution d'un problème multidimensionnel peut cependant donner également lieu à d'autres situations. Il se peut, notamment, qu'avec $\lambda = \bar{\lambda}$ l'hyperplan de la forme linéaire soit parallèle à la face Γ d'un ensemble convexe des conditions de dimension supérieure à l'unité. Dans ce cas, la variation ultérieure de λ n'aboutira pas forcément à un sommet de l'ensemble convexe voisin de A. De cette façon, deux sommets de l'ensemble convexe associés aux solutions du problème relatives à deux intervalles voisins de variation du paramètre λ ne sont absolument pas astreints à être voisins.

Dans le cas où les coefficients de la forme linéaire dépendent linéairement du paramètre λ, une modification négligeable de la méthode du simplexe permet de construire successivement en un nombre fini d'itérations les programmes optimaux pour toute la marge de variation du paramètre λ et de faire ressortir le domaine (s'il existe) où le problème est insoluble parce que la forme linéaire est infinie.

1.2. Considérons le problème paramétrique suivant. Déterminer pour chacun des λ d'un certain ensemble le vecteur X_λ qui maximise la forme linéaire

$$L = \sum_{j=1}^{n} c_j x_j = \sum_{j=1}^{n} (c_j' + \lambda c_j'') x_j \tag{1.1}$$

sous les conditions

$$\sum_{j=1}^{n} a_{ij} x_j = b_i, \quad i = 1, 2, \ldots, m; \tag{1.2}$$

$$x_j \geqslant 0, \quad j = 1, 2, \ldots, n. \tag{1.3}$$

Nous dirons que la base du programme de base du problème (1.1)-(1.3) est *optimale* pour un certain λ si les estimations par rapport à cette base de tous les vecteurs des conditions, calculées avec λ donné, sont non négatives. L'ensemble complet des valeurs du paramètre λ pour lesquelles la base est optimale s'appelle *ensemble d'optimalité* de cette base.

Examinons comment se modifie la solution du problème (1.1)-(1.3) avec la variation du paramètre λ. Utilisons n'importe quel algorithme de la méthode du simplexe et cherchons la solution du problème pour un certain $\lambda = \lambda_0$. Après un nombre fini d'étapes nous aboutirons au programme optimal (cas 1°) ou à la forme linéaire (1.1) infinie sur l'ensemble des programmes du problème (cas 2°). Considérons chacun de ces cas isolément.

(Le cas où les conditions (1.2)-(1.3) sont incompatibles ne présente pas d'intérêt, du fait que, quel que soit λ, le problème n'a pas de programmes.)

1°. Calculons les estimations $\Delta_j(\lambda)$ des vecteurs A_j par rapport à la base optimale trouvée $A_{i_1}, \ldots, A_{i_m}$ (pour $\lambda = \lambda_0$) du problème

$$\Delta_j(\lambda) = \sum_{s=1}^{m} c_{i_s}(\lambda)\, x_{i_s j} - c_j(\lambda) = \sum_{s=1}^{m} c'_{i_s} x_{i_s j} - c'_j + \lambda \Big[\sum_{s=1}^{m} c''_{i_s} x_{i_s j} - c''_j\Big].$$

Introduisons la notation:

$$\Delta_{j1} = \sum_{s=1}^{m} c'_{i_s} x_{i_s j} - c'_j,$$

$$\Delta_{j2} = \sum_{s=1}^{m} c''_{i_s} x_{i_s j} - c''_j.$$

Il vient

$$\Delta_j(\lambda) = \Delta_{j1} + \lambda \Delta_{j2}.$$

La base étant optimale pour $\lambda = \lambda_0$,

$$\Delta_j(\lambda_0) \geqslant 0, \quad j = 1, 2, \ldots, n,$$

c'est-à-dire que le système d'inégalités

$$\Delta_{j1} + \lambda \Delta_{j2} \geqslant 0, \quad j = 1, 2, \ldots, n \tag{1.4}$$

est compatible. Pour tout $\Delta_{j2} < 0$, les inégalités du système peuvent être mises sous la forme

$$\lambda \leqslant -\frac{\Delta_{j1}}{\Delta_{j2}},$$

et pour tout $\Delta_{j2} > 0$, on a

$$\lambda \geqslant -\frac{\Delta_{j1}}{\Delta_{j2}}.$$

Introduisons la notation :

$$\underline{\lambda} = \begin{cases} \max\limits_{\Delta_{j2}>0} \left(-\dfrac{\Delta_{j1}}{\Delta_{j2}}\right), \\ -\infty \text{ si tous les } \Delta_{j2} \leqslant 0 ; \end{cases}$$
$$\overline{\lambda} = \begin{cases} \min\limits_{\Delta_{j2}<0} \left(-\dfrac{\Delta_{j1}}{\Delta_{j2}}\right), \\ \infty \text{ si tous les } \Delta_{j2} \geqslant 0. \end{cases} \tag{1.5}$$

Par définition, la base considérée $A_{i_1}, A_{i_2}, \ldots, A_{i_m}$ est optimale pour celles et seulement celles des valeurs de λ qui vérifient le système d'inégalités (1.4).

Par conséquent, l'ensemble d'optimalité de la base considérée se compose de toutes les valeurs finies de λ qui vérifient la condition

$$\underline{\lambda} \leqslant \lambda \leqslant \overline{\lambda}.$$

Cette conclusion peut s'énoncer sous la forme de la proposition suivante.

Théorème 1.1. *Les nombres $\underline{\lambda}$ et $\overline{\lambda}$ caractérisent entièrement l'ensemble d'optimalité $E(\underline{\lambda}, \overline{\lambda})$ de la base donnée :*

si $\underline{\lambda} > \overline{\lambda}$, $E(\underline{\lambda}, \overline{\lambda})$ est un ensemble vide ;

si $\underline{\lambda} = \overline{\lambda}$, $E(\underline{\lambda}, \overline{\lambda})$ est le point $\lambda = \underline{\lambda} = \overline{\lambda}$;

si $-\infty < \underline{\lambda} < \overline{\lambda} < \infty$, $E(\underline{\lambda}, \overline{\lambda})$ est le segment $[\underline{\lambda}, \overline{\lambda}]$;

si $\underline{\lambda} = -\infty$, $\overline{\lambda} < \infty$, $E(\underline{\lambda}, \overline{\lambda})$ est la demi-droite $[-\infty, \overline{\lambda}]$;

si $-\infty < \underline{\lambda}$, $\overline{\lambda} = \infty$, $E(\underline{\lambda}, \overline{\lambda})$ est la demi-droite $[\underline{\lambda}, \infty]$;

si $\underline{\lambda} = -\infty$, $\overline{\lambda} = \infty$, $E(\underline{\lambda}, \overline{\lambda})$ est toute la droite.

Nous laissons au lecteur le soin de vérifier que chacune des éventualités indiquées par le théorème est effectivement réalisable (cf. exercice 1).

Passons maintenant à l'exploration du problème (1.1)-(1.3) pour les valeurs de $\lambda > \overline{\lambda}$. Il est logique de considérer ici que $\overline{\lambda} < \infty$, c'est-à-dire de supposer que parmi les nombres Δ_{j2} il en est qui sont négatifs. Soit k l'un des indices définis par la relation

$$\overline{\lambda} = \min_{\Delta_{j2}<0} \left(-\frac{\Delta_{j1}}{\Delta_{j2}}\right) = -\frac{\Delta_{k1}}{\Delta_{k2}}, \quad \Delta_{k2} < 0. \tag{1.6}$$

Il est clair que $\Delta_k(\overline{\lambda}) = 0$. Examinons les deux possibilités suivantes.

a) Toutes les composantes x_{ik} de la décomposition du vecteur des conditions A_k par rapport aux vecteurs de la base sont non posi-

tives. Sous cette condition, la forme linéaire du problème, pour $\lambda > \bar{\lambda}$, est infinie dans son domaine de définition. En effet, dans ce cas,

$$\Delta_k(\lambda) = \Delta_k(\bar{\lambda}) + (\lambda - \bar{\lambda})\,\Delta_{k2} = (\lambda - \bar{\lambda})\,\Delta_{k2} < 0$$
$$\text{pour } \lambda > \bar{\lambda}.$$

Cette inégalité, jointe à la condition $x_{ik} \leqslant 0$ pour tout i, est un critère d'absence de solution du problème pour la méthode primale du simplexe (cf. § 2 du chapitre 7 de [87]).

b) L'une au moins des composantes x_{ik} est positive. En poursuivant la résolution du problème par la méthode primale du simplexe, on introduit dans la base le vecteur A_k tel que

$$\Delta_k(\bar{\lambda}) = 0 \text{ et } \Delta_{k2} < 0,$$

et on élimine de la base le vecteur A_r tel que

$$\frac{x_{r0}}{x_{rk}} = \min_{x_{ik}>0} \frac{x_{i0}}{x_{ik}}, \quad x_{rk} > 0.$$

La nouvelle base est une base optimale au moins pour $\lambda = \bar{\lambda}$.

En effet, d'après les formules associant les paramètres des deux itérations successives de la méthode primale du simplexe

$$\Delta'_j(\lambda) = \Delta_j(\lambda) - \frac{x_{rj}}{x_{rk}}\,\Delta_k(\lambda), \; j = 1, 2, \ldots, n.$$

Pour $\lambda = \bar{\lambda}$, on a

$$\Delta'_j(\bar{\lambda}) = \Delta_j(\bar{\lambda}) \geqslant 0, \; j = 1, 2, \ldots, n.$$

Par conséquent avec $\lambda = \bar{\lambda}$, les inégalités

$$\Delta'_{j1} + \lambda\Delta'_{j2} \geqslant 0, \; j = 1, 2, \ldots, n \tag{1.7}$$

sont compatibles.

Les inégalités (1.7) peuvent être également compatibles avec $\lambda > \bar{\lambda}$; toutefois, le système (1.7) n'est pas satisfait par n'importe quel $\lambda < \bar{\lambda}$.

En effet, lors du passage à la nouvelle base, les paramètres Δ_{j1} et Δ_{j2} se transforment d'après les formules

$$\Delta'_{j1} = \Delta_{j1} - \frac{x_{rj}}{x_{rk}}\,\Delta_{k1},$$

$$\Delta'_{j2} = \Delta_{j2} - \frac{x_{rj}}{x_{rk}}\,\Delta_{k2}.$$

Le vecteur A_r appartient à la base. Par suite,

$$\Delta_{r1} = \Delta_{r2} = 0$$

et

$$\Delta'_{r1} = -\frac{\Delta_{k1}}{x_{rk}}, \quad \Delta'_{r2} = -\frac{\Delta_{k2}}{x_{rk}}. \tag{1.8}$$

Pour toutes les valeurs de λ qui vérifient (1.7) a lieu en particulier l'inégalité

$$\Delta'_{r1} + \lambda\Delta'_{r2} \geqslant 0,$$

ou, en raison de (1.8) et de la positivité de x_{rk},

$$\Delta_{k1} + \lambda\Delta_{k2} \leqslant 0.$$

Par construction $\Delta_{k2} < 0$. Aussi pour tout λ qui satisfait à la relation (1.7), on a l'inégalité

$$\lambda \geqslant -\frac{\Delta_{k1}}{\Delta_{k2}} = \bar{\lambda}.$$

Le résultat obtenu peut être énoncé sous la forme de la proposition suivante.

Théorème 1.2. *Soit $\bar{\lambda} < \infty$ et l'indice k défini par la condition* (1.6).

a) *Si $x_{ik} \leqslant 0$ ($i = 1, 2, \ldots, m$), pour tout $\lambda > \bar{\lambda}$ la forme linéaire* (1.1) *est infinie sur l'ensemble* (1.2)-(1.3).

b) *Si certains des nombres x_{ik} sont positifs, en introduisant dans la base disponible le vecteur A_k suivant les règles usuelles de la méthode primale du simplexe, on obtient une base nouvelle, dont l'extrémité gauche λ' de l'ensemble d'optimalité est $\bar{\lambda}$.*

Ainsi, la procédure de résolution du problème paramétrique pour $\lambda > \bar{\lambda}$ consiste à suivre les bases de ses programmes voisins, la borne droite de l'ensemble d'optimalité de la base précédente constituant la borne gauche de l'ensemble d'optimalité de la base suivante. La procédure s'interrompt sur l'obtention d'une demi-droite qui constitue soit l'ensemble d'optimalité de la dernière base, soit l'ensemble dans chaque point intérieur duquel le problème n'a pas de solution.

Il est aisé de s'assurer qu'en utilisant, pour choisir le vecteur à éliminer de la base, la règle de la méthode du simplexe qui permet d'éviter le cyclage, nous suivrons les bases différentes du problème.

Supposons, en effet, que deux bases B_s et B_{s+l}, obtenues au cours de la discussion du problème, coïncident. Soient $\underline{\lambda}$ et $\bar{\lambda}$ les bornes inférieure et supérieure de l'ensemble d'optimalité de la base $B \equiv \equiv B_s \equiv B_{s+l}$. La proposition b) du théorème 1.2 entraîne que

$$\underline{\lambda} \geqslant \bar{\lambda},$$

ce qui signifie que l'ensemble d'optimalité de la base B est constitué d'un point unique, $\tilde{\lambda} = \underline{\lambda} = \bar{\lambda}$, d'où il résulte que $\tilde{\lambda}$ est également ensemble d'optimalité pour chacune des bases intermédiaires

$$B_{s+1}, \ldots, B_{s+l-1}.$$

C'est pourquoi les conditions

$$\Delta_k^{(s+t)}(\tilde{\lambda})=0, \quad \Delta_{k2}^{(s+t)}<0$$

constituent le critère de sélection du A_k à inclure dans la base B_{s+t} $(0 \leqslant t \leqslant l-1)$.

Ainsi, le déplacement suivant les bases $B_s, \ldots, B_{s+l}$ s'effectue selon les règles de la méthode primale du simplexe, appliquée au problème auxiliaire de maximisation de la forme linéaire

$$\sum_{j=1}^{n} c_j'' x_j \tag{1.9}$$

sous les conditions (1.2)-(1.3) et sous la contrainte supplémentaire

$$x_j = 0 \text{ si } j \notin J, \tag{1.10}$$

où J est l'ensemble des indices j tels que

$$\Delta_j^{(s)}(\tilde{\lambda}) = 0.$$

Etant donné ensuite que la méthode primale du simplexe rend le cyclage impossible, on aboutit à la condition

$$B_s \neq B_{s+l},$$

qui contredit notre hypothèse et démontre par là même la proposition énoncée.

Ainsi, au cours de la discussion d'un problème paramétrique, le retour à une base déjà dépassée est impossible. Par conséquent, la procédure toute entière trouve place en un nombre fini d'itérations.

Nous avons considéré jusqu'à présent le problème pour des valeurs de λ situées à droite de $\bar{\lambda}$, borne droite de l'ensemble d'optimalité de la base initiale.

La discussion du problème (1.1)-(1.3) pour $\lambda < \underline{\lambda}$ se conduit de façon analogue. La différence consiste seulement en ce que la relation (1.6) est remplacée par

$$\underline{\lambda} = \max_{\Delta_{j2}>0} \left(-\frac{\Delta_{j1}}{\Delta_{j2}}\right) = -\frac{\Delta_{k1}}{\Delta_{k2}}, \quad \Delta_{k2}>0. \tag{1.11}$$

L'exploration du cas 1° est ainsi achevée.

1.3. Il reste à examiner le cas 2°, où la procédure de résolution du problème (1.1)-(1.3) pour $\lambda = \lambda_0$ se termine par la construction d'une base vérifiée par les conditions d'absence de solution réalisable du problème

$$\Delta_k = \Delta_{kl} + \lambda_0 \Delta_{k2} < 0,$$
$$x_{ik} \leqslant 0 \text{ pour } i = 1, 2, \ldots, m, \tag{1.12}$$

où A_k est un certain vecteur qui n'entre pas dans la base.

Pour $\Delta_{k2} = 0$, les relations (1.12) sont respectées quelle que soit la valeur du paramètre; le problème (1.1)-(1.3) n'a pas de solution sur tout l'axe des λ. Si $\Delta_{k2} > 0$, les conditions (1.12) sont observées pour toutes les valeurs de

$$\lambda < \lambda_1 = -\frac{\Delta_{k1}}{\Delta_{k2}};$$

si par contre $\Delta_{k2} < 0$, ces conditions sont vérifiées pour tout $\lambda > \lambda_1$. Donc, pour $\Delta_{k2} > 0$, le problème (1.1)-(1.3) n'a pas de solution à gauche de λ_1, et avec $\Delta_{k2} < 0$ il ne l'a pas à droite de λ_1. Posons, pour fixer les idées, $\Delta_{k2} > 0$ (la deuxième possibilité demande un examen analogue). L'analyse du problème paramétrique sur la demi-droite $\lambda \geqslant \lambda_1$ débute par la résolution du problème pour $\lambda = \lambda_1$ en partant de la base disponible. Si la procédure s'achève par l'obtention d'une base optimale, l'étude ultérieure s'effectue conformément aux recommandations du cas 1° données ci-dessus. Mais si cette procédure prend fin en révélant l'absence de solution du problème:

$$\Delta'_s(\lambda_1) = \Delta'_{s1} + \lambda_1 \Delta'_{s2} < 0, \tag{1.13}$$

$$x'_{is} \leqslant 0 \text{ pour } i = 1, 2, \ldots, m,$$

la marche de l'analyse ultérieure dépend du signe de Δ'_{s2}. Dans le cas où $\Delta'_{s2} < 0$, les conditions (1.13) définissant l'absence de solution restent valides pour tout

$$\lambda \geqslant \lambda_2 = -\frac{\Delta'_{s1}}{\Delta'_{s2}}.$$

Compte tenu du fait que dans le cas présent λ_1 est plus grand que λ_2 et que le problème paramétrique, comme on l'avait établi auparavant, n'a pas de solution à gauche de λ_1, on aboutit à la conclusion que le problème étudié est insoluble pour toutes les valeurs du paramètre λ. Cette même conclusion est justifiée évidemment pour $\Delta'_{s2} = 0$.

Si par contre $\Delta'_{s2} > 0$, les conditions d'absence de solution (1.13) restent vraies pour tout $\lambda < \lambda_2$, et de plus, $\lambda_2 > \lambda_1$. Lorsque cette éventualité se réalise, on en tire la conclusion que le problème (1.1)-(1.3) n'a pas de solution à gauche de λ_2 et l'on passe à l'examen du cas $\lambda = \lambda_2$. La marche à suivre ultérieurement est déterminée par les règles déjà décrites, à cette différence près que λ_1 est remplacé par λ_2. En un nombre fini d'étapes (cf. exercice 2), on établit que le problème (1.1)-(1.3) est irréalisable sur toute la droite λ, ou bien, ayant obtenu la base optimale pour une certaine valeur du paramètre λ, on se retrouve dans les conditions du cas 1°.

Ainsi, la méthode décrite permet dans n'importe quel cas de poursuivre jusqu'au bout l'exploration du problème paramétrique en un nombre fini d'étapes.

1.4. Voici quelques remarques au sujet de l'algorithme de la méthode décrite de programmation linéaire.

La solution du problème est recherchée d'après les règles générales du premier ou du second algorithme de la méthode primale du simplexe (cf. [87], chapitre 8). Indiquons seulement ici certaines particularités de la procédure de calcul, associées à la programmation paramétrique. Au lieu d'une ligne d'estimations des vecteurs des conditions [$(m+1)$-ième ligne des tableaux du premier algorithme ou ligne Δ du tableau auxiliaire du deuxième algorithme] on introduit trois lignes Δ_{j1}, Δ_{j2} et $-\frac{\Delta_{j1}}{\Delta_{j2}}$ pour le cas 1° et deux lignes $\Delta_{j1}+$ $+\lambda_s\Delta_{j2}$ et Δ_{j2} pour le cas 2°. (Pour le calcul du programme de base initial, bien entendu, les lignes supplémentaires ne sont pas nécessaires.)

La procédure de résolution du problème paramétrique débute par sa discussion (suivant les règles générales de la méthode du simplexe) pour une certaine valeur de $\lambda = \lambda_0$. L'analyse se termine par le cas 1° ou le cas 2°.

Après une mise en évidence du cas 1°, on introduit, comme on l'a déjà dit, trois lignes d'estimations. De plus, dans la ligne $-\frac{\Delta_{j1}}{\Delta_{j2}}$ on ne remplit que les cases qui correspondent à $\Delta_{j2} < 0$.

Si toutes les cases de la dernière ligne restent vides, la base courante est optimale pour tout $\lambda \geqslant \lambda_0$. S'il n'en est pas ainsi, l'indice de l'élément minimal de cette ligne détermine le numéro du vecteur à inclure dans la base, et la valeur de cet élément minimal coïncide avec la borne droite de l'ensemble d'optimalité de la base courante.

Si l'étude préalable du problème aboutit au cas 2°, $\Delta_{k2} \leqslant 0$ définit l'absence de solution pour tout $\lambda \geqslant \lambda_0$, et $\Delta_{k2} > 0$ impose de remplir les lignes

$$\Delta_{j1}+\lambda_1\Delta_{j2}\ \left(\lambda_1 = -\frac{\Delta_{k1}}{\Delta_{k2}}\right) \text{ et } \Delta_{j2}.$$

Le choix du vecteur à ajouter à la base est déterminé par l'un des éléments négatifs de la première ligne.

Nous nous sommes bornés ici à un examen sommaire de l'algorithme adapté à l'exploration du problème paramétrique pour $\lambda \geqslant \lambda_0$. Si la nécessité se présente d'étudier le problème à gauche du point λ_0, il faut modifier l'algorithme en apportant des corrections naturelles déterminées par le contenu des articles précédents : dans le cas 1°, on ne remplit dans la ligne $-\frac{\Delta_{j1}}{\Delta_{j2}}$ que les cases qui correspondent à $\Delta_{j2} > 0$, le vecteur à introduire dans la base étant déterminé alors par l'élément maximal de cette ligne ; dans le cas 2°, la procédure n'est poursuivie qu'avec $\Delta_{k2} < 0$ (la condition $\Delta_{k2} \geqslant 0$ traduit l'absence de solution à gauche du point λ_0).

Si l'analyse préalable du problème conduit au cas 2°, on aboutit au bout d'un nombre fini d'étapes à une solution irréalisable du problème (à gauche ou à droite de λ_0), ou bien l'on passe au cas 1°. L'étude du problème d'après les règles du cas 1° s'achève en un nombre fini d'itérations soit par la construction d'une base dont l'ensemble d'optimalité est une demi-droite, soit par la mise en évidence d'une demi-droite à l'intérieur de laquelle le problème n'a pas de solution (la direction de la demi-droite correspond à celle de la modification du paramètre lors de l'analyse du problème).

Au cours de la résolution du problème paramétrique il est plus avantageux de se déplacer dans une direction fixée. Si la marge imposée de variation du paramètre λ est une demi-droite ou un segment, on a donc intérêt de commencer les calculs par l'une de ses extrémités. Si par contre le problème doit être exploré sur tout l'axe λ, on peut choisir pour origine $\lambda = \infty$ ou $\lambda = -\infty$ (dans le premier cas, l'étude préalable impose la maximisation de la forme linéaire

$$\sum_{j=1}^{n} c_j'' x_j ;$$

dans le deuxième, elle se ramène à la minimisation de cette forme).

Pour finir, insistons sur la conclusion suivante tirée du contenu du présent paragraphe.

Dans le cas $C = C' + \lambda C''$, la procédure de calcul de la programmation paramétrique peut être considérée comme celle de résolution du problème par la méthode primale du simplexe avec l'application d'une règle particulière de choix du vecteur à ajouter à la base et des critères spéciaux pour mettre fin aux calculs.

1.5. Considérons un exemple. Maximiser pour toutes les valeurs de λ la forme linéaire

$$L_\lambda (X) = (2 + 3\lambda) x_1 + (-1 + 2\lambda) x_2 + 3\lambda x_3 + 4x_4$$

sous les conditions

$$\begin{aligned} x_1 + 2x_2 + x_3 + 3x_4 &\leqslant 7, \\ -3x_1 + 4x_2 + 3x_3 - x_4 &\leqslant 15, \\ 2x_1 - 5x_2 + 2x_3 + 2x_4 &\leqslant 2, \\ x_j \geqslant 0,\ j = 1,\ 2,\ 3,\ 4. \end{aligned}$$

Introduisons les variables non négatives additionnelles x_5, x_6 et x_7 et ramenons les conditions-inégalités à des égalités. Commençons la résolution du problème à partir de la base initiale A_5, A_6, A_7.

Le problème est résolu par le premier algorithme de la méthode primale du simplexe. La procédure est portée sur le tableau 3.1. Trois itérations seulement conduisent au programme optimal du problème avec $\lambda = 0$. Les lignes Δ_{j2} et $\left(-\frac{\Delta_{j1}}{\Delta_{j2}}\right)$ ne deviennent nécessaires qu'à la troisième itération, lorsqu'on a obtenu la base optimale pour $\lambda = 0$.

Tableau 3.1

—	—	c_j / C_x	—	—	$2+3\lambda$	$1+2\lambda$	3λ	4				—
	N		B	A_0	A_1	A_2	A_3	A_4	A_5	A_6	A_7	θ
	1		A_5	7	1	2	1	3	1			7/3
	2		A_6	15	—3	4	3	—1		1		—
←	3		A_7	2	2	—5	2	2			1	$\boxed{1}$
	4	—	Δ_{j1}		—2	1		$\boxed{-4}$				—
←	1		A_5	4	—2	19/2	—2		1		—3/2	$\boxed{8/19}$
	2		A_6	16	—2	3/2	4			1	1/2	32/3
→	3	4	A_4	1	1	—5/2	1	1			1/2	—
	4	—	Δ_{j1}	4	2	$\boxed{-9}$	4				2	—
⇄	1	$-1+2\lambda$	A_2	8/19	—4/19	1	—4/19		2/19		—3/19	$\boxed{4}$
	2		A_6	292/19	—32/19		82/19		—3/19	1	14/19	—
←	3	4	A_4	39/19	2/19		9/19	1	5/19		2/19	39/5

	4	—	Δ_{j1}	148/19	2/19		40/19		18/19		11/19
	5	—	Δ_{j2}	16/19	—65/19		—65/19		4/19		—6/19
	6	—	$-\Delta_{j1}/\Delta_{j2}$	—	$\boxed{2/65}$		8/13		$\boxed{-9/2}$		11/6
→	1		A_5	4	—2	19/2	—2		1		—3/2
	2		A_6	16	—2	3/2	4			1	1/2
	3	4	A_4	1	1	—5/2	1	1			1/2
	4	—	Δ_{j1}	4	2	—9	4				2
	5	—	Δ_{j2}		—3	—2	—3				
	6	—	$-\Delta_{j1}/\Delta_{j2}$	—							
	1	$-1+2\lambda$	A_2	4/3		1		4/9	2/9		—1/9
	2		A_6	68/3			6	32/9	7/9	1	10/9
→	3	$2+3\lambda$	A_1	13/3	1		1	19/9	5/9		2/9
	4	—	Δ_{j1}	22/3			2	—2/9	8/9		5/9
	5	—	Δ_{j2}	47/3				65/9	19/9		4/9
	6	—	$-\Delta_{j1}/\Delta_{j2}$	—							

Dans la ligne $\left(-\frac{\Delta_{j1}}{\Delta_{j2}}\right)$, le plus petit élément répondant à des valeurs négatives de Δ_{j2} est égal à 2/65 et correspond au vecteur des conditions A_1, et le plus grand élément répondant à des valeurs positives de Δ_{j2} est égal à $-9/2$ et correspond au vecteur A_5.

Par conséquent, une base, optimale avec $\lambda = 0$, reste encore une base de la solution du problème sur le segment $[-9/2, 2/65]$. Pour se déplacer à gauche le long de l'axe λ, il faut introduire dans la base le vecteur A_5, et pour se déplacer à droite, il faut introduire le vecteur A_1. Ajoutons d'abord à la base le vecteur A_5. On voit dès la première itération que tous les Δ_{j2} sont non positifs. Cela signifie que la base A_5, A_6, A_4, optimale avec $\lambda = -9/2$, constitue encore une base de la solution pour tous les points de la demi-droite $[-\infty, -9/2]$. Introduisons maintenant dans la base de la solution correspondant au segment $-9/2 \leqslant \lambda \leqslant 2/65$ le vecteur des conditions A_1. Dès la première itération, tout Δ_{j2} devient non négatif. Cela signifie que la base A_2, A_6, A_1, optimale avec $\lambda = 2/65$, constitue encore une base de la solution pour tout λ appartenant à la demi-droite $[2/65, \infty]$.

Ainsi, pour le problème considéré, l'axe λ peut être divisé en trois parties. Pour les points de la demi-droite $[-\infty, -9/2]$, la base de la solution est composée des vecteurs A_4, A_5, A_6, le programme optimal est égal à $X = (0, 0, 0, 1, 4, 16, 0)$, et le maximum de la forme linéaire est égal à 4. Sur le segment $[-9/2, 2/65]$, la base optimale est constituée des vecteurs A_2, A_4, A_6, la solution est égale à $X = (0, 8/19, 0, 39/19, 0, 292/19, 0)$ et le maximum de la forme linéaire est égal à $\frac{148 + 16\lambda}{19}$. Enfin pour les points de la demi-droite $[2/65, \infty]$, la base de la solution est composée des vecteurs A_1, A_2, A_6, le programme optimal est égal à $X = (13/3, 4/3, 0, 0, 0, 68/3, 0)$ et la valeur maximale de la forme linéaire est égale à $\frac{22 + 47\lambda}{3}$.

1.6. Comme nous l'avons montré aux paragraphes précédents, dans le cas où le problème (1.1)-(1.3) est résoluble pour plus d'une valeur du paramètre λ, l'axe λ tout entier peut être divisé en segments

$$[\lambda_{-t}, \lambda_{-(t-1)}], [\lambda_{-(t-1)}, \lambda_{-(t-2)}], \ldots, [\lambda_{-1}, \lambda_1], \ldots$$
$$\ldots, [\lambda_{r-2}, \lambda_{r-1}], [\lambda_{r-1}, \lambda_r]$$

et en demi-droites

$$(-\infty, \lambda_{-t}], [\lambda_r, \infty),$$

qui jouissent des propriétés suivantes:

chaque segment est un ensemble d'optimalité d'une certaine base;

chaque demi-droite est un ensemble d'optimalité d'une base, ou bien est composée de points λ (à l'exception de l'extrémité) pour lesquels la forme linéaire (1.1) n'est pas bornée supérieurement sur l'ensemble convexe (1.2)-(1.3).

Soit λ_s la frontière entre deux ensembles d'optimalité voisins (deux segments ou un segment et une demi-droite) dont celui de gauche correspond à la base B_s et celui de droite à la base B_{s+1}. D'après l'algorithme de programmation paramétrique décrit ci-dessus, le passage de B_s à B_{s+1} s'effectue généralement en plusieurs itérations, l'ensemble d'optimalité de chaque base intermédiaire

étant le point $\lambda = \lambda_s$. Il résulte de la règle de choix du vecteur à inclure dans la base que le processus de déplacement de B_s vers B_{s+1} est équivalent à la procédure de résolution du problème auxiliaire qui consiste à maximiser la forme linéaire (1.9) avec les conditions (1.2), (1.3) et la contrainte supplémentaire (1.10) (nous avons déjà utilisé cette circonstance pour prouver que l'algorithme est fini). L'interprétation géométrique de l'ensemble des conditions du problème auxiliaire est donnée par la face de l'ensemble connexe (1.2)-(1.3) qui est balayée par la forme linéaire (1.1) lorsque $\lambda = \lambda_s$.

Introduisons une fonction $f(\lambda)$ pour ceux des points λ pour lesquels le problème (1.1)-(1.3) a une solution réalisable en l'égalisant à la valeur maximale de la forme linéaire (1.1) sous les conditions (1.2)-(1.3).

Il est clair que le domaine de définition E de la fonction $f(\lambda)$ est soit une droite, soit une demi-droite ($[\lambda_{-t}, \infty)$ ou $(-\infty, \lambda_r]$), soit un segment $[\lambda_{-t}, \lambda_r]$.

Soit X_s le programme de base associé à la base B_s. Par définition, X_s est une solution du problème (1.1)-(1.3) pour tout λ appartenant à l'ensemble d'optimalité E_s de la base B_s. Par conséquent,

$$f(\lambda) = L'(X_s) + \lambda L''(X_s),$$

si $\lambda \in E_s$, où

$$L'(X) = \sum_{j=1}^{n} c'_j x_j, \quad L''(X) = \sum_{j=1}^{n} c''_j x_j.$$

Puisqu'au cours du passage de X_s à X_{s+1} la forme linéaire $L''(X)$ du problème auxiliaire est maximisée,

$$L''(X_s) \leqslant L''(X_{s+1}). \tag{1.14}$$

Si la relation (1.14) est vérifiée en tant qu'égalité stricte, cela signifie qu'à l'étape donnée seule a changé la base du programme. Aussi l'égalité

$$L''(X_s) = L''(X_{s+1})$$

entraîne la coïncidence des programmes X_s et X_{s+1}, bien que, bien entendu, les bases B_s et B_{s+1} soient différentes (ceci n'est naturellement possible que pour un programme de base dégénéré $X_s = X_{s+1}$). Ainsi, le changement du programme optimal en point λ_s a lieu si et seulement si

$$L''(X_s) < L''(X_{s+1}). \tag{1.15}$$

Le point λ_s pour lequel est vérifiée l'inégalité (1.15) est dit *critique*.

Soit

$$\lambda_{s_1}, \lambda_{s_2}, \ldots, \lambda_{s_k}$$

la collection complète des points critiques. Pour tout $\lambda \in [\lambda_{s_i}, \lambda_{s_{i+1}}]$, la solution du problème (1.1)-(1.3) est constituée par un même programme X_i. Il en va de même pour la demi-droite $(-\infty, \lambda_{s_1}]$ (ou $(\lambda_{s_k}, \infty]$) si elle n'est pas l'ensemble des solutions irréalisables du problème (nous désignerons les programmes correspondants par X_0 et X_k). Constatons qu'avec $i < j$,

$$L''(X_i) < L''(X_j).$$

Par conséquent, les segments et les demi-droites formés par les points critiques λ_{s_i} qui divisent l'axe λ constituent les ensembles d'optimalité complets pour les programmes X_i correspondants. Il est utile de formuler les conclusions obtenues sous la forme d'une proposition unique.

Théorème 1.3. *Si le problème* (1.1)-(1.3) *a une solution réalisable pour plus d'une valeur de* λ :

a) *le domaine de définition* E *de la fonction* $f(\lambda)$ *est soit une droite, soit une demi-droite, soit un segment*; $f(\lambda)$ *est continue sur* E;

b) *les points critiques* λ_{s_i} *(s'il y en a) divisent* E *en ensembles* E_{s_i} *(segments ou demi-droites) sur chacun desquels* $f(\lambda)$ *est une fonction linéaire*;

c) *un ensemble* E_{s_i} *quelconque est constitué des ensembles d'optimalité des bases d'un même programme de base* X_i:

d) *lorsqu'on se déplace de gauche à droite par les points critiques, les coefficients angulaires de la fonction linéaire par morceaux* $f(\lambda)$ *croissent monotonement.*

Les points critiques λ_{s_i} et les programmes de base X_i, qui déterminent parfaitement la fonction $f(\lambda)$, se calculent successivement au cours de l'exploration du problème (1.1)-(1.3) à l'aide de l'algorithme de programmation paramétrique décrit. Il résulte de la proposition d) du théorème 1.3 que $f(\lambda)$ est une fonction convexe vers le bas (cf. exercice 4).

§ 2. Cas $B = B' + \mu B''$

2.1. La figure 3.2 représente pour le cas le plus simple ($m = 2$) la deuxième interprétation géométrique de la résolution du problème de programmation linéaire. Son programme optimal répond au point supérieur M d'intersection de la droite Q et du cône polyédrique K dont les arêtes correspondent aux vecteurs complets des conditions du problème.

Ajoutons au vecteur des contraintes B' le vecteur $\mu B''$. Le déplacement du point d'intersection de la droite Q et du plan u_1Ou_2 provoque le déplacement du point supérieur d'intersection de la droite Q et du cône polyédrique K.

Si le point M de la face AOF du cône K ne repose pas sur une arête du cône (c'est-à-dire si le programme de base correspondant du problème est non dégénéré), μ n'étant pas grand, le point supérieur $M(\mu)$ d'intersection de $Q(\mu)$ et de K reste sur la même face. La base du programme optimal du problème, définie par les vecteurs complets dirigés le long de OA et OF, ne change pas. Une augmentation ultérieure de μ peut amener le point supérieur d'intersection de la droite

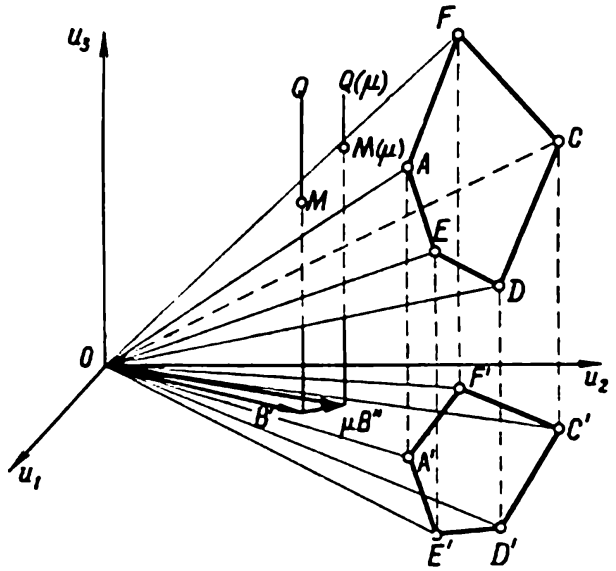

Fig. 3.2

$Q(\mu)$ et du cône K sur une arête du cône, le long de laquelle la face AOF est coupée par l'une des faces voisines. La solution du problème correspondant sera un programme dégénéré. Une variation ultérieure de μ transférera le point $M(\mu)$ sur la face voisine du cône K. La base optimale pour la marge suivante de variation de μ différera de la base de la solution du problème pour le segment précédent de la variation de μ par un vecteur des conditions.

En poursuivant ainsi la procédure de la variation du paramètre, on obtient les segments $[\mu_i, \mu_{i+1}]$ dont chacun est associé à la même base optimale du problème. Il se peut que lorsque μ varie, le vecteur $B(\mu) = B' + \mu B''$ sorte de l'angle $F'OE'$ formé par les vecteurs extrêmes des conditions. La droite Q correspondante ne coupe pas le cône polyédrique K. Le paramètre μ se trouve dans le domaine des solutions irréalisables du problème.

Le passage du cas $m = 2$ au cas $m > 2$ fait perdre la compréhension immédiate de l'image géométrique, mais garde les conclusions qualitatives précitées. Les constructions analytiques correspondantes indiquent la procédure à suivre pour couper en un nombre fini d'étapes la marge des variations de μ en un nombre fini de secteurs sur chacun desquels la base optimale ne varie pas avec la varia-

tion de μ et dégager le domaine (s'il existe) dans lequel la solution du problème est irréalisable.

Remarquons que, comme dans le paragraphe précédent, l'interprétation géométrique du problème pour $m = 2$ ne permet pas d'interpréter le cas où le passage à la base optimale pour la marge successive de variations du paramètre s'effectue non pas en une, mais en plusieurs itérations.

2.2. Considérons le problème de programmation linéaire paramétrique qui consiste à calculer, pour tout μ appartenant à un certain ensemble de l'axe réel, un vecteur X tel qu'il maximise la forme linéaire

$$\sum_{j=1}^{n} c_j x_j \tag{2.1}$$

sous les conditions

$$\sum_{j=1}^{n} a_{ij} x_j = b_i' + \mu b_i'', \; i = 1, 2, \ldots, m; \tag{2.2}$$

$$x_j \geqslant 0, \; j = 1, 2, \ldots, n. \tag{2.3}$$

Au lieu du problème (2.1)-(2.3), on aurait pu, certes, examiner son dual (dans lequel le paramètre μ ferait partie de la forme linéaire) et l'explorer en faisant appel aux conclusions du § 1. Toutefois, nous préférons recourir ici à une analyse indépendante du problème (2.1)-(2.3), fondée sur la méthode duale du simplexe.

De même que dans la méthode duale du simplexe, la procédure de calcul consiste à se déplacer suivant les bases des pseudoprogrammes du problème (2.1)-(2.3).

Rappelons qu'on appelle base d'un pseudoprogramme ou *pseudobase* un système de m vecteurs des conditions linéairement indépendants par rapport auquel les critères de sélection de tous les vecteurs des conditions du problème sont non négatifs. Si, pour une certaine valeur du paramètre μ, les coefficients de décomposition du vecteur des contraintes du problème (2.1)-(2.3) selon les vecteurs de la pseudobase sont non négatifs, cette pseudobase constitue alors la base optimale du problème pour μ considéré. Par analogie avec le paragraphe précédent, appelons *ensemble d'optimalité* d'une pseudobase l'ensemble complet des valeurs du paramètre μ telles que la pseudobase considérée soit optimale.

Etudions le problème (2.1)-(2.3) pour $\mu = \mu_0$ en nous guidant par les règles de la méthode duale du simplexe. Finalement, on aura construit la base optimale pour $\mu = \mu_0$ (cas 1°), ou bien l'on aura mis en évidence l'incompatibilité des conditions (2.2)-(2.3) pour $\mu = \mu_0$ (cas 2°).

Le cas où, la forme linéaire étant infinie, le problème n'a pas de solution réalisable ne présente aucun intérêt. Le problème est alors

insoluble avec tout μ, les conditions du problème dual ne dépendant pas de μ et par conséquent étant incompatibles avec tout μ.

Considérons d'abord le cas 1°.

1°. Supposons que la base optimale obtenue se compose des vecteurs $A_{i_1}, \ldots, A_{i_m}$. On a

$$B' + \mu_0 B'' = \sum_{s=1}^{m} A_{i_s} x_{i_s}(\mu_0). \tag{2.4}$$

En résolvant le système (2.4) par rapport à $x_{i_s}(\mu_0)$, il vient

$$x_{i_s}(\mu_0) = x_{i_s}^{(1)} + \mu_0 x_{i_s}^{(2)} = x_{s0}^{(1)} + \mu_0 x_{s0}^{(2)}, \quad s = 1, 2, \ldots, m,$$

où $x_{i_s}^{(1)} = x_{s0}^{(1)}$ $(s = 1, 2, \ldots, m)$ est la solution du système (2.4) pour $B = B'$; $x_{i_s}^{(2)} = x_{s0}^{(2)}$ $(s = 1, 2, \ldots, m)$, la solution de ce même système pour $B = B''$.

Toutes les composantes de base $x_{i_s}(\mu_0) = x_{s0}(\mu_0)$ sont non négatives. Cela signifie que le système d'inégalités

$$x_{s0}(\mu) = x_{s0}^{(1)} + \mu x_{s0}^{(2)} \geqslant 0, \; s = 1, 2, \ldots, m \tag{2.5}$$

est compatible.

Les critères Δ_j des vecteurs A_j ne dépendent pas de μ. C'est pourquoi une condition nécessaire et suffisante d'optimalité d'une pseudobase est que les inégalités (2.5) soient vérifiées. Considérons le système (2.5).

La relation relative au numéro s du système (2.5) est vérifiée si avec $x_{s0}^{(2)} > 0$

$$\mu \geqslant -\frac{x_{s0}^{(1)}}{x_{s0}^{(2)}},$$

et avec $x_{s0}^{(2)} < 0$

$$\mu \leqslant -\frac{x_{s0}^{(1)}}{x_{s0}^{(2)}}.$$

Introduisons les notations

$$\underline{\mu} = \begin{cases} \max\limits_{x_{s0}^{(2)} > 0} \left(-\dfrac{x_{s0}^{(1)}}{x_{s0}^{(2)}}\right), \\ -\infty \text{ si tous les } x_{s0}^{(2)} \leqslant 0; \end{cases} \tag{2.6}$$

$$\overline{\mu} = \begin{cases} \min\limits_{x_{s0}^{(2)} < 0} \left(-\dfrac{x_{s0}^{(1)}}{x_{s0}^{(2)}}\right), \\ \infty \text{ si tous les } x_{s0}^{(2)} \geqslant 0. \end{cases}$$

Le système des inégalités (2.5) étant résoluble, $\underline{\mu} \leqslant \overline{\mu}$.

La pseudobase considérée est optimale pour celles et seulement pour celles des valeurs de μ qui sont vérifiées par le système d'inégalités (2.5). Par suite, l'ensemble d'optimalité de cette pseudobase se compose de toutes sortes de valeurs finies de μ qui vérifient la

condition

$$\underline{\mu} \leqslant \mu \leqslant \overline{\mu}.$$

On obtient ainsi la proposition suivante.

Théorème 2.1. *L'ensemble d'optimalité* $E(\underline{\mu}, \overline{\mu})$ *de la base considérée est bien défini par les nombres* $\underline{\mu}$ et $\overline{\mu}$ *calculés d'après les formules* (2.6) :

si $\underline{\mu} > \overline{\mu}$, $E(\underline{\mu}, \overline{\mu})$ *est un ensemble vide*;

si $\underline{\mu} = \overline{\mu}$, $E(\underline{\mu}, \overline{\mu})$ *est le point* $\mu = \underline{\mu} = \overline{\mu}$;

si $-\infty < \underline{\mu} < \overline{\mu} < \infty$, $E(\underline{\mu}, \overline{\mu})$ *est le segment* $[\underline{\mu}, \overline{\mu}]$;

si $\underline{\mu} = -\infty$, $\overline{\mu} < \infty$, $E(\underline{\mu}, \overline{\mu})$ *est la demi-droite* $(-\infty, \overline{\mu}]$;

si $\underline{\mu} > -\infty$, $\overline{\mu} = \infty$, $E(\underline{\mu}, \overline{\mu})$ *est la demi-droite* $[\underline{\mu}, \infty)$;

si $\underline{\mu} = -\infty$, $\overline{\mu} = \infty$, $E(\underline{\mu}, \overline{\mu})$ *est toute la droite.*

Il est facile de s'assurer que chacune des 6 éventualités du théorème 2.1 peut avoir lieu (cf. exercice 5). Examinons maintenant le problème (2.1)-(2.3) à droite du point $\mu = \overline{\mu}$ (pour $\mu < \underline{\mu}$ la procédure est analogue). Tout ceci n'a naturellement de sens que si $\overline{\mu} < \infty$, c'est-à-dire si parmi les $x_{s0}^{(2)}$ il y a des nombres négatifs. Pour tout $\mu \in E(\underline{\mu}, \overline{\mu})$, le pseudoprogramme $X(\mu)$ à base $A_{i_1}, A_{i_2}, \ldots, A_{i_m}$ est solution du problème (2.1)-(2.3). Avec $\mu > \overline{\mu}$, une ou plusieurs composantes de base du pseudoprogramme $X(\mu)$ peuvent devenir négatives. Soit $x_{r0}(\mu)$ la première des composantes du pseudoprogramme qui change de signe. Pour elle, $x_{r0}^{(2)} < 0$ et

$$\overline{\mu} = \min_{x_{i0}^{(2)} < 0} \left(-\frac{x_{i0}^{(1)}}{x_{i0}^{(2)}} \right) = -\frac{x_{r0}^{(1)}}{x_{r0}^{(2)}}. \tag{2.7}$$

Avec $\mu > \overline{\mu}$, les vecteurs $A_{i_1}, \ldots, A_{i_m}$ ne forment déjà plus une base optimale, bien que le vecteur $X(\mu)$ ne cesse pas d'être pseudoprogramme du problème (du fait que $\Delta_j \geqslant 0$ indépendamment de la valeur du paramètre μ).

Effectuons une transformation élémentaire du pseudoprogramme $X(\mu)$. D'après les règles de la méthode duale du simplexe éliminons de la base le vecteur A_{i_r} pour le remplacer par le vecteur A_k tel que

$$-\frac{\Delta_k}{x_{rk}} = \min_{x_{rj} < 0} \left(-\frac{\Delta_j}{x_{rj}} \right). \tag{2.8}$$

On suppose ici qu'il existe au moins un vecteur A_j tel que $x_{rj} < 0$. S'il n'en est pas ainsi, c'est-à-dire si $x_{rj} \geqslant 0$ avec $j = 1, 2, \ldots, n$, les principes théoriques de la méthode duale du simplexe (cf. [87], § 1, chapitre 9) entraînent qu'avec $\mu > \overline{\mu}$ le problème étudié est irréalisable (puisque $x_{r0}(\mu) < 0$ avec $\mu > \overline{\mu}$).

Les composantes de base des pseudoprogrammes obtenus l'un à partir de l'autre par des transformations élémentaires sont liées par les relations de récurrence

$$\left.\begin{aligned} x'_{i0}(\mu) &= x_{i0}^{(1)\prime} + \mu x_{i0}^{(2)\prime} = \\ &= x_{i0}^{(1)} + \mu x_{i0}^{(2)} - \frac{x_{ik}}{x_{rk}}(x_{r0}^{(1)} + \mu x_{r0}^{(2)}) \ (i \neq r), \\ x'_{r0}(\mu) &= x_{r0}^{(1)\prime} + \mu x_{r0}^{(2)\prime} = \frac{x_{r0}^{(1)} + \mu x_{r0}^{(2)}}{x_{rk}}. \end{aligned}\right\} \tag{2.9}$$

Les critères Δ_j des vecteurs des conditions sont recalculés d'après les formules analogues:

$$\Delta'_j = \Delta_j - \frac{x_{rj}}{x_{rk}}\Delta_k. \tag{2.10}$$

Montrons que la nouvelle pseudobase est optimale avec $\mu = \bar{\mu}$ et peut l'être avec $\mu > \bar{\mu}$, mais ne le peut pas avec $\mu < \bar{\mu}$. En effet, il suit de (2.7) et (2.9) que

$$x'_{i0}(\bar{\mu}) = x_{i0}(\bar{\mu}) \geqslant 0.$$

On tire de (2.9) que le pseudoprogramme $X'(\mu)$ ne peut être un programme du problème que si

$$\frac{x'_{r0} + \mu x_{r0}^{(2)}}{x_{rk}} \geqslant 0.$$

Mais $x_{r0}^{(2)}) < 0$, $x_{rk} < 0$. Donc, $X'(\mu)$ ne peut être un programme du problème qu'avec

$$\mu \geqslant -\frac{x_{r0}^{(1)}}{x_{r0}^{(2)}} = \bar{\mu}.$$

Les conclusions ainsi obtenues peuvent être énoncées sous la forme de la proposition suivante.

Théorème 2.2. *Soit* $\bar{\mu} < \infty$ *et l'indice* r *défini par la relation* (2.7).

a) Si $x_{rj} \geqslant 0$ *pour* $j = 1, 2, \ldots, n$, *avec* $\mu > \bar{\mu}$ *les conditions* (2.2)-(2.3) *du problème sont incompatibles*;

b) Si certains des nombres x_{rj} $(j = 1, 2, \ldots, n)$ *sont négatifs, en éliminant de la pseudobase le vecteur* A_{i_r}, *d'après les règles de la méthode duale du simplexe, on obtient une pseudobase nouvelle dont l'extrémité gauche* μ' *de l'ensemble d'optimalité est égale à* $\bar{\mu}$.

Ainsi, la procédure du problème paramétrique avec $\mu > \bar{\mu}$ consiste à se déplacer suivant des pseudobases voisines de ce problème. Par ailleurs, l'extrémité droite de l'ensemble d'optimalité de la pseudobase précédente coïncide avec l'extrémité gauche de l'ensemble d'optimalité de la pseudobase suivante.

La procédure s'achève par la construction de la demi-droite qui constitue soit l'ensemble d'optimalité de la dernière base, soit l'ensemble tel que les conditions du problème sont incompatibles en tout point intérieur.

On peut montrer que si, pour définir l'indice k du vecteur à inclure dans la base, on utilise la règle qui assure l'absence de cyclage dans la méthode duale du simplexe, on se déplacera toujours en suivant des pseudobases différentes. Les raisonnements respectifs sont analogues à ceux du paragraphe 1.2 relatifs au cas du paramètre entrant dans la forme linéaire du problème. Nous laissons au lecteur d'effectuer ces raisonnements lui-même (cf. exercice 6). Ainsi, tout retour à une pseudobase déjà passée est exclu, et par suite, la procédure considérée ne dépasse pas un nombre fini d'étapes.

2.3. Il reste à examiner le cas 2°, lorsque la résolution du problème (2.1)-(2.3) avec $\mu = \mu_0$ montre que ses conditions sont incompatibles.

2°. D'après le critère du cas 2°, il y a parmi les composantes $x_{i0}(\mu_0) = x_{i0}^{(1)} + \mu_0 x_{i0}^{(2)}$ des valeurs négatives, et pour l'une d'entre elles, au moins, tout $x_{ij} \geqslant 0$.

Soit

$$x_{r0}(\mu_0) = x_{r0}^{(1)} + \mu_0 x_{r0}^{(2)} < 0; \tag{2.11}$$

et, de plus, tout

$$x_{rj} \geqslant 0.$$

Si $x_{r0}^{(2)} = 0$, les conditions d'incompatibilité sont évidemment remplies pour toutes les valeurs de μ ; le problème (2.1)-(2.3) est partout irréalisable. Avec $x_{r0}^{(2)} > 0$,

$$x_{r0}(\mu) < 0 \quad \text{pour} \quad \mu < -\frac{x_{r0}^{(1)}}{x_{r0}^{(2)}} = \mu_1,$$

c'est-à-dire que le problème est insoluble à gauche du point μ_1.

Avec $x_{r0}^{(2)} < 0$,

$$x_{r0}(\mu) < 0 \text{ pour } \mu > \mu_1,$$

les conditions du problème sont incompatibles à droite de μ_1.

Supposons pour fixer les idées que $x_{r0}^{(2)} > 0$ et explorons le problème (2.1)-(2.3) sur la demi-droite $[\mu_1, \infty)$. Appliquons au problème paramétrique avec $\mu = \mu_1$ la méthode duale du simplexe en partant de la pseudobase disponible. Après quelques itérations, on obtient un programme optimal, ou bien l'on constate que le problème est impossible

$$\begin{gathered} \bar{x}_{i0}(\mu_1) = \bar{x}_{i0}^{(1)} + \mu_1 \bar{x}_{i0}^{(2)} < 0, \\ \bar{x}_{ij} \geqslant 0 \text{ pour } j = 1, 2, \ldots, n. \end{gathered} \tag{2.12}$$

La première éventualité nous amène au cas 1° ; la discussion ultérieure se poursuit suivant les recommandations du paragraphe 2.2. Si, par contre, on est en présence des relations (2.12), celles-ci, avec $\bar{x}_{i0}^{(2)} \leqslant 0$, restent évidemment en vigueur pour tout $\mu \geqslant \mu_1$,

alors qu'avec $\bar{x}_{i0}^{(2)} > 0$, elles restent vraies à gauche de

$$\mu_2 = -\frac{\bar{x}_{i0}^{(1)}}{\bar{x}_{i0}^{(2)}}.$$

Par conséquent, la non-positivité de $\bar{x}_{i0}^{(2)}$ signifie que le problème (2.1)-(2.3) est insoluble sur tout l'axe des μ, et la positivité de ce paramètre indique que le problème est irréalisable à gauche du point $\mu_2 > \mu_1$. Dans ce dernier cas, l'exploration du problème paramétrique se poursuit conformément aux règles déjà exposées, à cette différence près que μ_1 est remplacé par μ_2. Après un nombre fini d'étapes on aboutit au cas 1°, ou bien on établit que les conditions (2.2)-(2.3) sont incompatibles pour toutes les valeurs du paramètre μ. La démonstration de ce fait est laissée au lecteur (cf. exercice 6).

2.4. Choisissons un algorithme de programmation paramétrique pour le cas où le vecteur des contraintes dépend linéairement du paramètre.

La solution du problème débute par le calcul de sa base optimale pour l'un des points μ_0 (généralement arbitraire) de la marge donnée de variation du paramètre μ. Si cette marge est une demi-droite ou un segment, il est commode d'aborder le calcul par le point extrême, pour avancer pendant la résolution du problème toujours dans le même sens.

La solution du problème pour le point initial μ_0 est recherchée par la méthode primale ou duale du simplexe suivant qu'il est plus simple de calculer le programme de base initial ou le pseudoprogramme du problème.

Si au terme de la discussion du problème (2.1)-(2.3) avec $\mu = \mu_0$ il apparaît que sa forme linéaire est infinie, on en conclut que la solution de ce problème est irréalisable pour toutes les valeurs de μ. Si par contre la discussion s'achève par la construction d'un programme optimal (cas 1°) ou par la mise en évidence de l'incompatibilité de ses conditions (cas 2°), il faut utiliser la méthode de programmation paramétrique décrite dans les paragraphes précédents. L'algorithme de cette méthode s'obtient en modifiant légèrement l'un des algorithmes de la méthode duale du simplexe (le premier ou le deuxième). Dans le cas 1°, on aura intérêt à remplacer la colonne A_0 du tableau de la méthode duale du simplexe par trois colonnes, où figurent dans la première (A_0') les composantes $x_{i0}^{(1)}$, dans la deuxième (A_0'') les composantes $x_{i0}^{(2)}$, dans la troisième les rapports $-\frac{x_{i0}^{(1)}}{x_{i0}^{(2)}}$. Si au cours de l'étude le paramètre μ croît (décroît), on ne complète alors que celles des cases de la troisième colonne auxquelles correspondent les $x_{i0}^{(2)} < 0$ ($x_{i0}^{(2)} > 0$).

Le vecteur à éliminer de la base est déterminé par l'élément minimal de la troisième colonne lorsque μ croît, et par l'élément maximal de cette colonne lorsque μ décroît. Pour le reste, l'itération est identique à une étape isolée de la méthode duale du simplexe. En cas de croissance (de décroissance) du paramètre μ, l'élément minimal (maximal) de la troisième colonne définit l'extrémité droite (gauche) de l'ensemble d'optimalité de la pseudobase courante.

La procédure prend fin soit en rendant évidentes les conditions d'irrésolubilité

$$x_{rj} \leqslant 0, \ j = 1, 2, \ldots, n$$

(r est le numéro du vecteur à retrancher de la base), soit avec l'absence totale d'éléments dans la troisième colonne. Dans le premier cas, on note que le problème est insoluble à droite du point μ^* si au cours de la procédure le paramètre μ croît et à gauche du point μ^* si μ décroît. μ^* est ici l'élément de la troisième colonne qui correspond au vecteur à éliminer de la base. Dans le deuxième cas, la pseudobase courante est optimale pour la demi-droite $[\mu^*, \infty)$ si μ croît, et pour la demi-droite $(-\infty, \mu^*]$ si μ décroît.

Considérons le cas 2°, où l'étude préalable du problème s'est achevée par la mise en évidence de l'incompatibilité de ses conditions avec $\mu = \mu_0$.

Soit r le numéro de la ligne qui vérifie les conditions d'incompatibilité :

$$x_{r0}(\mu_0) = x_{r0}^{(1)} + \mu_0 x_{r0}^{(2)} < 0, \ x_{rj} \geqslant 0, \ j = 1, 2, \ldots, n.$$

Avec $x_{r0}^{(2)} = 0$, on conclut que les conditions du problème sont incompatibles pour tout μ. Dans le cas contraire, on calcule $\mu_1 = -\frac{x_{r0}^{(1)}}{x_{r0}^{(2)}}$ et on remplace dans les tableaux la colonne A_0 par deux colonnes où figurent dans la première (A_0^*) les composantes $x_{i0}^{(2)}$, et dans la deuxième ($A_0(\mu)$) les composantes $x_{i0}(\mu_1) = x_{i0}^{(1)} + \mu_1 x_{i0}^{(2)}$.

Si $x_{r0}^{(2)} > 0$, le problème (2.1)-(2.3) n'a pas de solution réalisable à gauche de μ_1, et l'analyse doit être poursuivie seulement pour les points de la demi-droite $[\mu_1, \infty]$. Si $x_{r0}^{(2)} < 0$, l'exploration du problème doit concerner seulement la demi-droite $(-\infty, \mu_1]$: à droite du point μ_1 les conditions du problème sont incompatibles. Le choix du vecteur à éliminer de la base est guidé par l'un des éléments négatifs de la deuxième colonne. Pour le reste, on applique toujours les règles de la méthode duale du simplexe. Si la procédure s'achève par la construction d'un programme optimal, l'étude se poursuit d'après les recommandations du cas 1°. Mais si l'on met en évidence l'absence de solution réalisable (le r_1-ième élément de la deuxième colonne est négatif, les éléments de la r_1-ième ligne à partir du troisième sont non négatifs), la marche ultérieure du calcul est déterminée par le signe du r_1-ième élément de la colonne A_0^*.

Supposons que l'étude du problème se fasse à droite (à gauche) du point μ^*. Avec $\bar{x}_{r10}^{(2)} \leqslant 0$ ($\bar{x}_{r10}^{(2)} \geqslant 0$), on note que les conditions du problème sont incompatibles sur tout l'axe des μ. Avec $\bar{x}_{r10}^{(2)} > 0$ ($\bar{x}_{r10}^{(2)} < 0$), la colonne A_0 (μ_1) est remplacée par A_0 (μ_2), où

$$\mu_2 = -\frac{\bar{x}_{r_1 0}^{(1)}}{\bar{x}_{r_1 0}^{(2)}},$$

et la procédure d'exploration du problème paramétrique se poursuit selon les recommandations décrites ci-dessus pour le cas 2°. De plus, à gauche (à droite) de μ_2, on fixe l'incompatibilité des conditions du problème. Le contenu du présent paragraphe permet de tirer la conclusion suivante.

Dans le cas $B = B' + \mu B''$, de la programmation paramétrique la procédure de calcul consiste à résoudre le problème par la méthode duale du simplexe en appliquant une règle particulière de choix du vecteur à éliminer de la base et des critères spéciaux pour mettre fin aux calculs.

2.5. Considérons un exemple à titre d'illustration.

Maximiser pour toutes les valeurs du paramètre μ la forme linéaire

$$L = -x_1 + 2x_2 - 3x_3 - x_4$$

sous les conditions

$$\begin{aligned} 2x_1 + x_2 - 3x_3 + x_4 &\leqslant 1 - 2\mu, \\ x_1 + 3x_2 + 4x_3 + 2x_4 &\leqslant 20 + \mu, \\ -4x_1 + 5x_2 - 2x_3 + 3x_4 &\leqslant 5 + 3\mu, \\ x_j \geqslant 0,\ j = 1,\ 2,\ 3,\ 4. \end{aligned}$$

Introduisons les variables supplémentaires non négatives x_5, x_6, x_7 et ramenons aux égalités les conditions-inégalités.

Pour $\mu = 0$, le programme de base initial est évident. Aussi est-il plus commode de calculer la base de la solution du problème à l'aide de la méthode primale du simplexe. Le passage de la base initiale A_5, A_6, A_7 à la base optimale pour $\mu = 0$, (A_2, A_6, A_3), se fait en trois itérations. Le vecteur des contraintes se décompose par rapport aux bases des programmes de base en complétant les colonnes A_0' et A_0''. A chaque itération consécutive, les composantes $x_{i0}^{(2)}$ se calculent à partir des valeurs correspondantes obtenues à l'itération précédente suivant les mêmes formules de récurrence que celles utilisées pour le calcul de $x_{i0}^{(1)}$. La colonne $\left(-\frac{x_{i0}^{(1)}}{x_{i0}^{(2)}}\right)$ ne se complète qu'après le calcul de la base optimale pour $\mu = 0$.

La procédure de résolution du problème, qui utilise le premier algorithme, est donnée dans le tableau 3.2.

La troisième itération établit la base optimale du problème pour $\mu = 0$. Les coefficients de décomposition du vecteur des contraintes par rapport à la base A_2, A_6, A_3 sont égaux respectivement à $x_{10} = 1 + \mu$, $x_{20} = 17 - 6\mu$, $x_{30} = \mu$. L'élément maximal de la colonne $\left(-\frac{x_{i0}^{(1)}}{x_{i0}^{(2)}}\right)$ avec $x_{i0}^{(2)} > 0$ est égal à 0 et s'obtient dans la case qui correspond au vecteur A_3 de la base.

Tableau 3.2

—	—	c_j	—	—	—	—	−1	2	−3	−1				—
	N	C_X	B_X	A'_0	A''_0	$-\dfrac{x_{i_0}^{(1)}}{x_{i_0}^{(2)}}$	A_1	A_2	A_3	A_4	A_5	A_6	A_7	0
←	1		A_5	1	−2	—	2	1	−3	1	1			$\boxed{1}$
	2		A_6	20	1	—	1	3	4	2		1		20/3
	3		A_7	5	3	—	−4	5	−2	3			1	1
	4	—	Δ			—	1	$\boxed{-2}$	3	1				—
→	1	2	A_2	1	−2	—	2	1	−3	1	1			—
	2		A_6	17	7	—	−5		13	−1	−3	1		17/3
←	3		A_7		13	—	−14		13	−2	−5		1	$\boxed{0}$
	4	—	Δ	2	−4	—	5		$\boxed{-3}$	3	2			—
	1	2	A_2	1	1	−1	−16/13	1		7/13	−2/13		3/13	
←	2		A_6	17	−6	$\boxed{17/6}$	9			1	2	1	−1	
⇄	3	−3	A_3		1	$\boxed{0}$	−14/13		1	−2/13	−5/13		1/13	
	4	—	Δ	2	−1	—	23/13			33/13	11/13		3/13	

	5	—	$-\frac{\Delta_j}{x_{rj}}$			—	$\boxed{23/14}$	—	—	33/2	11/5	—	—
	1	2	A_2	1	—1/7	—		1	—8/7	5/7	2/7		1/7
←	2		A_6	17	33/14	—238/33			117/14	—2/7	—17/14	1	—5/14
→	3	—1	A_1		—13/14	—	1		—13/14	1/7	5/14		—1/14
	4	—	Δ	2	9/14				23/14	16/7	3/14		5/14
	5	—	$-\frac{\Delta_j}{x_{rj}}$			—	—	—	—	8	$\boxed{3/17}$	—	1
	1	2	A_2	5	7/17	$\boxed{-85/7}$		1	14/17	11/17		4/17	1/17
→	2		A_5	—14	—33/17	—			—117/17	4/17	1	—14/17	5/17
	3	—1	A_1	5	—4/17	—	1		26/17	1/17		5/17	—3/17
	4	—	Δ	5	18/17				53/17	38/17		3/17	5/17
	5	—	$-\frac{\Delta_j}{x_{rj}}$										
	1	2	A_2	64/13	—5/13	$\boxed{64/5}$	11/13	1		10/13	4/13	3/13	
→	2		A_7	—17	6	—	—9			—1	—2	—1	1
	3	—3	A_3	17/13	7/13	—	—5/13		1	—1/13	—3/13	1/13	
	4	—	Δ	77/13	—31/13		50/13			36/13	17/13	3/13	

Dans la colonne A_0^* le seul élément négatif correspond à la deuxième case de la base dans laquelle figure le vecteur A_6. Le rapport $-\frac{x_{20}^{(1)}}{x_{20}^{(2)}}$ est égal à $\frac{17}{6}$.

Dans la colonne $-\frac{x_{i0}^{(1)}}{x_{i0}^{(2)}}$ on a marqué deux éléments: $\mu_0 = 0$ et $\mu_1 = \frac{17}{6}$, qui déterminent le segment de l'axe μ sur lequel le système de vecteurs A_2, A_6, A_3 constitue la base de la solution. Le vecteur

$$X = (0,\ 1 + \mu,\ \mu,\ 0,\ 0,\ 17 - 6\mu,\ 0)$$

est un programme optimal, et la valeur maximale de la forme linéaire est

$$L^* = 2 - \mu.$$

Pour passer au demi-axe négatif μ il faut retrancher de la base le vecteur A_3, alors que l'exploration du problème avec $\mu > \mu_1 = \frac{17}{6}$ impose l'élimination du vecteur A_6.

Eliminons d'abord le vecteur A_3. La ligne $-\frac{\Delta_j}{x_{rj}}$ et les règles de la méthode duale du simplexe indiquent qu'il faut ajouter à la base le vecteur A_1. En réalisant l'itération suivante, seules sont complétées dans la colonne $-\frac{x_{i0}^{(1)}}{x_{i0}^{(2)}}$ les cases qui correspondent à $x_{i0}^{(2)} > 0$. La valeur positive de $x_{i0}^{(2)}$ n'est associée qu'au vecteur A_6, qui figure dans la troisième case. La valeur $-\frac{x_{30}^{(1)}}{x_{30}^{(2)}} = -\frac{238}{33} \approx$ $\approx -7,21$ détermine le point μ_{-1}. Sur le segment $[\mu_{-1}, \mu_0]$, la base de la solution du problème est constituée par le système de vecteurs A_2, A_6, A_1. Le programme optimal du problème est égal à

$$X = \left(-\frac{13}{14}\mu,\ 1 - \frac{1}{7}\mu,\ 0,\ 0,\ 0,\ 17 + \frac{33}{14}\mu,\ 0\right),$$

et la valeur maximale de la forme linéaire est

$$L^* = 2 + \frac{9}{14}\mu.$$

L'exploration de l'axe μ plus à gauche impose l'élimination de la base du vecteur A_6. Le vecteur à inclure dans la base est A_5, auquel correspond le plus petit élément de la ligne $-\frac{\Delta_j}{x_{rj}}$. Le seul élément de la colonne des rapports détermine le point $\mu_{-2} = -\frac{85}{7}$. Sur le segment $[\mu_{-2}, \mu_{-1}]$, le système de vecteurs A_2, A_5, A_1 constitue la base de la solution, le vecteur

$$X = \left(5 - \frac{4}{17}\mu,\ 5 + \frac{7}{17}\mu,\ 0,\ 0,\ -14 - \frac{33}{17}\mu,\ 0,\ 0\right)$$

est le programme optimal, et la valeur maximale de la forme linéaire est

$$L^* = 5 + \frac{18}{17}\mu.$$

La valeur μ_{-2} s'obtient dans la première case de la base, qui correspond au vecteur A_2. Les coefficients de décomposition x_{ij} de tous les vecteurs des conditions par rapport à la base sont non négatifs. Conformément au critère d'irré-

solubilité du problème, ceci signifie qu'avec $\mu < \mu_{-2}$ le problème n'a pas de programmes.

Il reste à explorer le problème pour $\mu > \mu_1 = \frac{17}{6}$. A cette fin reprenons la base A_2, A_6, A_3, optimale sur le segment $[\mu_0, \mu_1]$, éliminons le vecteur A_6 placé en deuxième case et remplaçons-le par le vecteur A_7 tel qu'il fournit $\min\limits_{x_{2j}<0} \left(-\frac{\Delta_j}{x_{2j}}\right)$.

Dans le tableau qui correspond à la nouvelle base (A_2, A_7, A_3), la valeur de $-\frac{x_{10}^{(1)}}{x_{10}^{(2)}}$ définit l'extrémité droite du segment d'optimalité successif $\mu_2 = \frac{64}{5}$. Sur le segment $[\mu_1, \mu_2] = \left[\frac{17}{6}, \frac{64}{5}\right]$, le système de vecteurs A_2, A_7, A_3 constitue la base de la solution, le vecteur

$$X = \left(0, \frac{64 - 5\mu}{13}, \frac{17 + 7\mu}{13}, 0, 0, 0, -17 + 6\mu\right)$$

est le programme optimal, et la valeur maximale de la forme linéaire est égale à

$$L^* = \frac{77 - 31\mu}{13}.$$

La valeur de μ_2 s'obtient dans la première case de la base qui correspond au vecteur A_2. Toutes les composantes x_{1j} de la décomposition des vecteurs des conditions du problème par rapport à la base sont non négatives. D'après le critère d'irrésolubilité du problème, ceci signifie qu'avec $\mu > \mu_2$ les conditions du problème sont incompatibles.

Le problème est ainsi étudié complètement. L'axe μ est divisé en quatre segments, sur chacun desquels la base de la solution ne change pas, et en deux demi-droites pour lesquelles le problème n'a pas de programmes.

Les systèmes de vecteurs explorés par la procédure de résolution du problème ont été appelés ici bases, puisqu'ils sont tels pour certaines valeurs du paramètre μ. Remarquons que ces systèmes sont tous des pseudobases pour n'importe quelle valeur de μ.

2.6. Au terme de la discussion ci-dessus du problème paramétrique (2.1)-(2.3), que l'on suppose résoluble pour plus d'une valeur de μ, l'axe μ est coupé en segments

$$[\mu_{-t}, \mu_{-(t-1)}], [\mu_{-(t-1)}, \mu_{-(t-2)}], \ldots, [\mu_{-1}, \mu_1], \ldots$$
$$\ldots, [\mu_{r-2}, \mu_{r-1}], [\mu_{r-1}, \mu_r]$$

et en demi-droites

$$(-\infty, \mu_{-t}], [\mu_r, \infty)$$

qui jouissent des propriétés suivantes:

tout segment est un ensemble d'optimalité pour une certaine pseudobase;

toute demi-droite est un ensemble d'optimalité d'une certaine pseudobase, ou bien se compose (extrémité exclue) des points μ pour lesquels les conditions (2.2)-(2.3) sont incompatibles.

Soit μ_s la frontière de deux ensembles d'optimalité voisins, B_s (B_{s+1}) la pseudobase associée à l'ensemble gauche (droit). Le passage de B_s à B_{s+1} s'effectue généralement en quelques itérations. Il est évident que chaque pseudobase intermédiaire compte un ensemble d'optimalité constitué par le point unique $\mu = \mu_s$.

Soit E l'ensemble des points μ tels que le problème (2.1)-(2.3) a une solution. Pour tout $\mu \in E$, calculons $f(\mu)$, le maximum de la forme linéaire (2.1), sous les conditions (2.2)-(2.3). Si E_s est l'ensemble d'optimalité de la pseudobase B_s, alors, pour $\mu \in E_s$

$$f(\mu) = \sum_{\alpha=1}^{m} c_{i_\alpha} (x_{\alpha 0}^{(1)} + \mu x_{\alpha 0}^{(2)}) = \delta_s^{(1)} + \mu \delta_s^{(2)},$$

où $x_{\alpha_0}^{(1)}$ ($x_{\alpha_0}^{(2)}$) sont les coefficients de décomposition du vecteur B' (B'') par rapport à la pseudobase $B_s = (A_{i_1}, A_{i_2}, \ldots, A_{i_m})$.

Par conséquent, $f(\mu)$ est une fonction linéaire de μ sur chacun des ensembles d'optimalité.

Disons que μ_s est un point critique si

$$\delta_s^{(2)} \neq \delta_{s+1}^{(2)}.$$

On a l'analogue suivant du théorème 1.3, dont la justification est laissée au lecteur (cf. exercice 8).

Théorème 2.3. *Si le problème* (2.1)-(2.3) *a une solution pour plus d'une valeur de* μ,

a) *le domaine de définition E de la fonction $f(\mu)$ est soit une droite, soit une demi-droite, soit un segment; $f(\mu)$ est continue sur E;*

b) *les points critiques μ_{s_i} (s'ils existent) divisent E en ensembles E_{s_i} (segments ou demi-droites) sur chacun desquels $f(\mu)$ est une fonction linéaire;*

c) *tout ensemble E_{s_i} est composé des ensembles d'optimalité des pseudobases aux mêmes critères de sélection;*

d) *les coefficients angulaires de la fonction linéaire par morceaux $f(\mu)$ décroissent monotonement en passant par les points critiques de gauche à droite, c'est-à-dire $f(\mu)$ est une fonction convexe vers le haut.*

§ 3. Cas général

3.1. Dans les paragraphes précédents nous avons étudié de près deux problèmes paramétriques simples dont l'exploration nécessite un nombre fini d'itérations de la méthode primale ou de la méthode duale du simplexe. Ce qu'il y a ici de commun, c'est que dans les deux cas l'axe du paramètre est balayé successivement par des ensembles connexes jointifs, dont chacun constitue par rapport au système donné de vecteurs des conditions soit l'ensemble d'optimalité pour une base fixe (pseudo-

base), soit l'ensemble de la forme linéaire infinie (des contraintes incompatibles). Le passage de l'ensemble disponible à l'ensemble successif nécessite une ou plusieurs itérations de la méthode correspondante. Rappelons également que dans les deux cas, les grandeurs x_{ij} et Δ_j associées à la base (pseudobase) fixe ne dépendent pas du paramètre, ou bien en dépendent linéairement. Cette circonstance simplifie l'algorithme de la méthode de programmation paramétrique.

Considérons le problème général de la programmation paramétrique linéaire, dans lequel les coefficients de la forme linéaire, les composantes des vecteurs des conditions et du vecteur des contraintes dépendent linéairement d'un paramètre.

Soit la forme linéaire à maximiser:

$$\sum_{j=1}^{n} (c_j' + tc_j'')\, x_j \tag{3.1}$$

sous les conditions

$$\sum_{j=1}^{n} (a_{ij}' + ta_{ij}'')\, x_j = b_i' + tb_i'', \qquad i = 1, 2, \ldots, m; \tag{3.2}$$

$$x_j \geqslant 0, \; j = 1, 2, \ldots, n. \tag{3.3}$$

L'analyse de ce problème, comme des problèmes plus simples examinés dans ce qui précède, se ramène également à balayer l'axe t par une collection finie d'ensembles (ensembles d'optimalité ou ensembles de solutions irréalisables de systèmes fixes de vecteurs des conditions). Toutefois, dans le cas présent, les ensembles qui balayent l'axe t ne sont pas astreints à être connexes, ce qui empêche le mouvement monotone le long de l'axe t et complique la discussion. En outre, le système de vecteurs des conditions du problème qui constitue sa base optimale dans le voisinage $(t_0 - \varepsilon, t_0)$ n'est pas astreint à être base ou pseudobase du problème à droite de t_0. Par conséquent, l'utilisation immédiate de la méthode primale ou de la méthode duale du simplexe est rendue malaisée. Enfin, dans la majorité des cas, les grandeurs Δ_j et x_{ij} sont des fonctions non linéaires du paramètre t. Ces circonstances (première et troisième surtout) rendent très complexe l'étude du problème général (3.1)-(3.3). Sous ce rapport, l'étude de certains cas particuliers du problème (3.1)-(3.3) est d'un grand intérêt pratique, leur allure singulière permettant d'obtenir des schémas de calcul relativement simples. Dans ce qui suit nous soumettons à un examen sommaire deux de ces cas. Les considérations qui y sont rapportées peuvent s'avérer utiles également lors de la discussion de situations plus complexes.

3.2. Supposons que $a_{ij}'' = 0$ pour tous i et j, c'est-à-dire que le paramètre t ne fait partie que des coefficients de la forme linéaire du problème (3.1)-(3.3) et des composantes de son vecteur des contraintes. Autrement dit, considérons le problème paramétrique qui

consiste à maximiser la forme linéaire

$$\sum_{j=1}^{n} (c'_j + tc''_j)\, x_j \tag{3.4}$$

sous les conditions

$$\sum_{j=1}^{n} A_j x_j = B' + tB'', \tag{3.5}$$

$$x_j \geqslant 0, \ j = 1, 2, \ldots, n. \tag{3.6}$$

Ici $A_j = (a_{1j}, \ldots, a_{mj})^T$, $B' = (b'_1, b'_2, \ldots, b'_m)^T$, $B'' = (b''_1, b''_2, \ldots, b''_m)^T$. Le problème (3.4)-(3.6) est une généralisation naturelle des problèmes étudiés aux §§ 1, 2.

Appelons *base fondamentale finie* tout système linéairement indépendant contenant m vecteurs des conditions du problème (3.4)-(3.6). L'ensemble des valeurs du paramètre t telles que la base fondamentale finie soit une base de la solution du problème (3.4)-(3.6) s'appelle *ensemble d'optimalité de la base*.

Soulignons qu'à la différence des problèmes des §§ 1, 2, dans le problème considéré la base fondamentale finie peut être une base pour certaines valeurs de t, une pseudobase pour d'autres valeurs, et pour d'autres valeurs encore peut ne posséder aucune de ces propriétés.

L'analyse du problème (3.4)-(3.6) débute par sa résolution pour une certaine valeur du paramètre $t = t_0$. La discussion préalable peut s'achever par l'un des trois résultats suivants:

a) construction d'une solution de base du problème à base

$$A_{i_1}, A_{i_2}, \ldots, A_{i_m};$$

b) mise en évidence de l'incompatibilité des conditions du problème;

c) mise en évidence du fait que la forme linéaire du problème est infinie sur l'ensemble de ses programmes.

Considérons seulement la première éventualité. Par x_{ij}, $x_{i0}^{(1)}$ et $x_{i0}^{(2)}$ nous désignerons, comme d'habitude, les coefficients de décomposition par rapport à la base des vecteurs A_j ($j = 1, 2, \ldots, n$), B' et B'' respectivement. Posons

$$\Delta_{j1} = \sum_{\alpha=1}^{m} c'_{i_\alpha} x_{\alpha j} - c'_j, \quad \Delta_{j2} = \sum_{\alpha=1}^{m} c''_{i_\alpha} x_{\alpha j} - c''_j, \quad j = 1, 2, \ldots, n.$$

Pour que le point t appartienne à l'ensemble d'optimalité de la base fondamentale finie $A_{i_1}, A_{i_2}, \ldots, A_{i_m}$, il faut et il suffit de vérifier les inégalités

$$x_{i0}(t) = x_{i0}^{(1)} + tx_{i0}^{(2)} \geqslant 0, \ i = 1, 2, \ldots, m; \tag{3.7}$$

$$\Delta_j(t) = \Delta_{j1} + t\Delta_{j2} \geqslant 0, \ j = 1, 2, \ldots, n. \tag{3.8}$$

Le système d'inégalités linéaires (3.7)-(3.8) est compatible, puisqu'il est vérifié par le point $t = t_0$.

Introduisons les quantités

$$\left.\begin{aligned} \underline{t}' &= \begin{cases} -\infty & \text{si } x_{i0}^{(2)} \leqslant 0 \text{ pour } i = 1, 2, \ldots, m; \\ \max\limits_{x_{i0}^{(2)} > 0} \left(-\dfrac{x_{i0}^{(1)}}{x_{i0}^{(2)}}\right) & \text{dans le cas contraire;} \end{cases} \\ \underline{t}'' &= \begin{cases} -\infty & \text{si } \Delta_{j2} \leqslant 0 \text{ pour } j = 1, 2, \ldots, n; \\ \max\limits_{\Delta_{j2} > 0} \left(-\dfrac{\Delta_{j1}}{\Delta_{j2}}\right) & \text{dans le cas contraire;} \end{cases} \end{aligned}\right\} \quad (3.9)$$

$$\left.\begin{aligned} \bar{t}' &= \begin{cases} \infty & \text{si } x_{i0}^{(2)} \geqslant 0 \text{ pour } i = 1, 2, \ldots, m; \\ \min\limits_{x_{i0}^{(2)} < 0} \left(-\dfrac{x_{i0}^{(1)}}{x_{i0}^{(2)}}\right) & \text{dans le cas contraire;} \end{cases} \\ \bar{t}'' &= \begin{cases} \infty & \text{si } \Delta_{j2} \geqslant 0 \text{ pour } j = 1, 2, \ldots, n; \\ \min\limits_{\Delta_{j2} < 0} \left(-\dfrac{\Delta_{j1}}{\Delta_{j2}}\right) & \text{dans le cas contraire.} \end{cases} \end{aligned}\right\} \quad (3.10)$$

L'ensemble des solutions du système d'inégalités (3.7)-(3.8) se compose des points extrémaux t qui obéissent à la condition

$$\underline{t} = \max(\underline{t}', \underline{t}'') \leqslant t \leqslant \min(\bar{t}', \bar{t}'') = \bar{t}. \qquad (3.11)$$

Ainsi, l'ensemble d'optimalité de la base fondamentale finie donnée est soit une droite ($\underline{t} = -\infty$, $\bar{t} = \infty$), soit une demi-droite (une des bornes $\underline{t}$, $\bar{t}$ est infinie, l'autre est finie), soit encore un segment ($\underline{t} > -\infty$, $\bar{t} < \infty$), qui peut dégénérer en un point ($\underline{t} = \bar{t} = t_0$). On voit que les ensembles d'optimalité du problème (3.4)-(3.6) ont la propriété d'être connexes du fait de la dépendance linéaire des grandeurs $x_{i0}(t)$ et $\Delta_j(t)$ par rapport au paramètre t. Cette circonstance, aussi bien que la linéarité des fonctions $x_{i0}(t)$ et $\Delta_j(t)$ simplifient considérablement la discussion du problème.

3.3. Supposons que la borne droite $\bar{t}$ de l'ensemble d'optimalité donné soit finie et essayons de poursuivre la discussion du problème pour $\underline{t} > \bar{t}$. Produisons un décalage suffisamment petit à droite de $\bar{t}$. Si

$$\bar{t} = \bar{t}' < \bar{t}'',$$

la base fondamentale $A_{i_1}, A_{i_2}, \ldots, A_{i_m}$ cesse alors d'être une base, mais reste encore une pseudobase, car les inégalités (3.8) continuent d'être vérifiées. Dans la situation donnée, il faut naturellement

utiliser, comme au § 2, la méthode duale du simplexe. Si, lors du passage par $\bar{t}$, une ou plusieurs inégalités du système (3.8) sont violées alors que le système (3.7) reste toujours vérifié, c'est-à-dire si

$$\bar{t} = \bar{t}'' < \bar{t}',$$

on recourt à la méthode primale du simplexe de même qu'il en a été au § 1.

On peut enfin avoir le cas limite où

$$\bar{t} = \bar{t}' = \bar{t}'',$$

que nous n'avions pas rencontré jusque-là. Quel que soit $t > \bar{t}$, la base fondamentale $A_{i_1}, A_{i_2}, \ldots, A_{i_m}$ n'est dans ce cas ni une base, ni une pseudobase. Afin de poursuivre l'étude du problème il faut faire appel aux considérations suivantes.

Remplaçons le vecteur des contraintes $B(t)$ du problème (3.4)-(3.6) par le vecteur $B(\bar{t})$ et appelons auxiliaire le problème ainsi obtenu. Il est clair que la base fondamentale $A_{i_1}, A_{i_2}, \ldots, A_{i_m}$ est une base du problème auxiliaire quelle que soit la valeur du paramètre t. C'est pourquoi, en partant de la base fondamentale disponible, on constate, après quelques itérations de la méthode primale du simplexe, que le problème est impossible avec $\bar{t} > t$ ou on construit une nouvelle base dont l'ensemble d'optimalité a pour extrémité gauche le point $\bar{t}$.

Le premier résultat indique que le problème paramétrique est insoluble pour $t > \bar{t}$, puisque les conditions des problèmes duals par rapport aux problèmes paramétrique et auxiliaire sont les mêmes. Le second résultat conduit à la base fondamentale qui est une pseudobase du problème (3.4)-(3.6) sur un certain segment à extrémité gauche en $\bar{t}$. Par conséquent, la discussion ultérieure peut se poursuivre par la méthode duale du simplexe en partant de la pseudobase nouvellement construite.

Ainsi, l'exploration du problème paramétrique (3.4)-(3.6) se ramène à une série d'étapes successives de la méthode primale et de la méthode duale du simplexe. La procédure est guidée par les règles déduites de ce qui précède. Bornons-nous au cas où le paramètre t varie de gauche à droite.

Calculons d'après la base fondamentale disponible les valeurs $\bar{t}'$ et $\bar{t}''$ en appliquant les formules (3.9) et (3.10). Si

$$\bar{t} = \min(\bar{t}', \bar{t}'') = \infty,$$

la discussion est terminée.

Mais si $\bar{t} < \infty$, il faut passer à l'itération suivante, dont le type est déterminé par la relation entre les nombres $\bar{t}'$ et $\bar{t}''$. Deux cas sont

possibles :

$$\text{a) } \bar{t}' < \bar{t}''; \quad \text{b) } \bar{t}' \geqslant \bar{t}''.$$

a) Dans le premier cas, on effectue une étape de la méthode duale du simplexe pour le problème (3.4)-(3.6) avec $t = \bar{t}$. Le vecteur A_{i_r} à retrancher de la base est défini par la relation

$$-\frac{x_{r0}^{(1)}}{x_{r0}^{(2)}} = \min_{x_{i0}^{(2)}<0} \left(-\frac{x_{i0}^{(1)}}{x_{i0}^{(2)}}\right) = \bar{t}, \quad x_{r0}^{(2)} < 0. \tag{3.11'}$$

b) Dans le deuxième cas, on effectue une étape de la méthode primale du simplexe pour le problème (3.4)-(3.6) avec $t = \bar{t}$. Le vecteur A_k à inclure dans la base est donné par la relation

$$-\frac{\Delta_{k1}}{\Delta_{k2}} = \min_{\Delta_{j2}<0} \left(-\frac{\Delta_{j1}}{\Delta_{j2}}\right) = \bar{t}, \quad \Delta_{k2} < 0. \tag{3.12}$$

S'il arrive que

$$x_{rj} \geqslant 0 \text{ pour } j = 1, 2, \ldots, n \text{ dans le cas a)}$$

ou

$$x_{ik} \leqslant 0 \text{ pour } i = 1, 2, \ldots, m \text{ dans le cas b),}$$

on note alors que le problème est irréalisable à droite de $\bar{t}$ [dans le premier cas l'incompatibilité porte sur les conditions du problème dual par rapport à (3.4)-(3.6), dans le deuxième, sur celles du problème (3.4)-(3.6) lui-même].

En appliquant les considérations exposées plus haut, il est aisé de voir que la procédure proposée permet d'assurer l'exploration complète du problème paramétrique (3.4)-(3.6) en un nombre fini d'étapes. Nous laissons au lecteur le soin d'effectuer les raisonnements correspondants (cf. exercice 11). L'algorithme que nous venons d'examiner permet de calculer successivement toutes les valeurs de la fonction

$$f(t) = \max_X \left[\sum_{j=1}^{n} (c_j' + tc_j'')\, x_j\right],$$

où le maximum est pris par rapport à X qui vérifient les conditions (3.5)-(3.6). Cet algorithme permet de démontrer sans peine la proposition suivante (exercice 10).

Théorème 3.1. *Supposons que le problème* (3.4)-(3.6) *ait une solution pour plus d'une valeur de* t. *Alors*:

a) le domaine de définition E *de la fonction* $f(t)$ *est soit une droite, soit une demi-droite, soit un segment*; $f(t)$ *est continue sur* E;

b) l'ensemble E *est coupé par un nombre fini de points* t_i *(points critiques) en ensembles* E_i *(segments ou demi-droites s'il y a au moins un point critique), sur chacun desquels* $f(t) = d_{i1}t^2 + d_{i2}t + d_{i3}$.

3.4. Considérons encore un cas particulier du problème (3.1)-(3.3) où le paramètre t n'entre que dans les composantes de l'un de ses vecteurs des conditions. Il s'agit du problème paramétrique qui consiste à maximiser la forme linéaire

$$\sum_{j=1}^{n} c_j x_j \tag{3.13}$$

sous les conditions

$$x_1 (A_1' + tA_1'') + \sum_{j=2}^{n} x_j A_j = B; \tag{3.14}$$

$$x_j \geqslant 0, \ j = 1, 2, \ldots, n. \tag{3.15}$$

Selon l'usage, l'exploration débute par l'analyse du problème pour une certaine valeur du paramètre $t = t_0$. L'étude préalable débouche sur l'un des trois résultats :

1) construction d'une base optimale composée de vecteurs d'indices $i_1, i_2, \ldots, i_m$;

2) mise en évidence de l'incompatibilité des conditions du problème ;

3) obtention d'une forme linéaire infinie sur l'ensemble des programmes du problème.

Nous n'examinerons ici qu'une seule éventualité, puisqu'un examen complémentaire de deux autres cas n'apporterait rien de nouveau en principe à l'algorithme de résolution du problème.

Supposons donc qu'on ait le cas 1). Calculons les coefficients de décomposition $x_{ij}(t)$ des vecteurs des conditions du problème et du vecteur $A_0 = B$ par rapport à la base disponible et composons les critères $\Delta_j(t) = x_{m+1,\, j}(t)$. Considérons par convention $i_s \geqslant 2$ avec $s = 2, 3, \ldots, m$. L'écriture des fonctions $x_{ij}(t)$ ($i = 1, 2, \ldots, m+1$; $j = 0, 1, \ldots, n$) varie suivant que le vecteur

$$A_1(t) = A_1' + A_1'' t$$

entre ou n'entre pas dans la base donnée.

Si $i_1 = 1$ et si le système de vecteurs $A_1'', A_{i_2}, \ldots, A_{i_m}$ est linéairement indépendant,

$$x_{ij}(t) = \begin{cases} x_{ij} \dfrac{t - t_{ij}}{t - t^*}, & i = 2, 3, \ldots, m+1; \ j = 0, 1, \ldots, n; \\ \dfrac{x_{ij}}{t - t^*}, & i = 1, \ j = 0, 1, \ldots, n. \end{cases} \tag{3.16}$$

Si $i_1 = 1$ et si le vecteur A_1'' est une combinaison linéaire des vecteurs $A_{i_2}, A_{i_3}, \ldots, A_{i_m}$,

$$x_{ij}(t) = x_{ij}^{(1)} + t x_{ij}^{(2)}, \quad i = 1, 2, \ldots, m+1; \quad j = 0, 1, \ldots n. \tag{3.17}$$

Si $i_1 > 1$,

$$x_{ij}(t) = \begin{cases} x_{ij}, & i = 1, 2, \ldots, m; \; j = 0, 2, \ldots, n; \\ x_{i1}^{(1)} + t x_{i1}^{(2)}, & i = 1, 2, \ldots, m; \; j = 1. \end{cases} \tag{3.18}$$

x_{ij}, t_{ij}, t^*, $x_{i1}^{(1)}$, $x_{i1}^{(2)}$ sont ici des constantes.

Les relations (3.16) et (3.17) sont une simple conséquence des formules bien connues de Cramer.

On voit que dans le cas considéré les éléments des tableaux de programmation linéaire peuvent dépendre non linéairement du paramètre t. Toutefois, le caractère spécifique de cette dépendance non linéaire [$x_{ij}(t)$ sont des fonctions homographiques de mêmes dénominateurs] permet quand même d'élaborer un algorithme assez économique de discussion du problème (3.13)-(3.15).

Déterminons le domaine d'optimalité de la base disponible.

Soit d'abord $i_1 = 1$ et le système de vecteurs A_1^*, A_{i_2}, ..., A_{i_m} linéairement indépendant. D'après (3.16) les conditions d'optimalité

$$x_{i0}(t) \geqslant 0, \quad \Delta_j(t) \geqslant 0$$

se mettent sous la forme

$$\frac{x_{10}}{t - t^*} \geqslant 0, \quad x_{i0} \frac{t - t_{i0}}{t - t^*} \geqslant 0, \quad i = 2, 3, \ldots, m; \tag{3.19}$$

$$x_{m+1,\,j} \frac{t - t_{m+1,\,j}}{t - t^*} \geqslant 0, \quad j = 1, 2, \ldots, n. \tag{3.20}$$

Si $x_{10} \neq 0$, le signe du dénominateur $t - t^*$ étant constant (il doit coïncider avec le signe de x_{10}), on aboutit à la conclusion que l'ensemble d'optimalité de la base fondamentale est connexe. Désignons par $\underline{t}$ et $\overline{t}$ les bornes inférieure et supérieure de l'ensemble d'optimalité (ces bornes se calculent sans peine). Pour t^* différent de $\underline{t}$ et $\overline{t}$, l'ensemble d'optimalité est fermé, c'est-à-dire coïncide soit avec une droite, soit avec une demi-droite, soit encore avec un segment qui peut dégénérer en un point. Lorsque t^* se confond avec l'un des nombres $\underline{t}$, $\overline{t}$, l'ensemble d'optimalité de la base fondamentale est soit une demi-droite, soit un segment privé d'une de ses extrémités.

Si $x_{10} = 0$, le vecteur B est une combinaison linéaire des vecteurs A_{i_2}, A_{i_3}, ..., A_{i_m}. Le cas le plus probable est que parmi les vecteurs des conditions du problème n'entrant pas dans la base fondamentale il se trouve au moins un vecteur A_j tel que $x_{1j}(t_0) \neq 0$. Dans ce cas, en remplaçant dans la base le vecteur $A_1(t)$ par A_j, nous ne changerons pas les composantes du programme de base disponible tout en amenant une situation qui répond à la formule (3.18). Il peut arriver toutefois que tout vecteur des conditions A_j, $j \geqslant 2$ soit une combinaison linéaire des vecteurs A_{i_2}, ..., A_{i_m}. Il est clair alors que les premiers membres des inégalités (3.19)-(3.20) ne dépendent

pas de t et que l'ensemble d'optimalité de la base donnée est une droite. Le cas relatif aux relations (3.16) est ainsi complètement étudié.

Les éventualités correspondant aux formules (3.17) et (3.18) sont plus simples, car dans ce cas la dépendance du paramètre est linéaire. Lors de la réalisation d'une de ces éventualités l'ensemble d'optimalité est toujours fermé et connexe.

Les conclusions obtenues peuvent être énoncées de la façon suivante.

Théorème 3.2. *Supposons que le système de vecteurs donné constitue une base du programme optimal du problème* (3.13)-(3.15) *pour une certaine valeur du paramètre t. L'ensemble d'optimalité de cette base est alors l'un des ensembles suivants: une droite, une demi-droite (fermée ou ouverte), un segment (fermé ou privé d'une de ses extrémités), ou un point.*

3.5. Voyons maintenant comment avancer, en partant d'une base fondamentale disponible, au-delà de l'une des bornes de son ensemble d'optimalité. Pour fixer les idées, nous parlerons de la borne droite $\bar{t}$, qu'il est logique dans ce cas de considérer comme finie.

Considérons le problème paramétrique (3.13)-(3.15) dans un intervalle suffisamment petit $(\bar{t}, \bar{t} + \varepsilon)$ de variation du paramètre t.

Trois cas sont possibles:

1) la base fondamentale disponible $S(t)$ est une base pour $t \in (\bar{t}, \bar{t} + \varepsilon)$;

2) la base fondamentale $S(t)$ est une pseudobase pour $t \in (\bar{t}, \bar{t} + \varepsilon)$;

3) pour $t \in (\bar{t}, \bar{t} + \varepsilon)$, aussi bien parmi les quantités $x_{i0}(t)$ $(1 \leqslant i \leqslant m)$ que parmi les quantités $\Delta_j(t) = x_{m+1,\, j}(t)$ $(1 \leqslant j \leqslant n)$, il en est qui sont négatives.

Dans le cas 1), (2)), on applique la méthode primale (duale) du simplexe en partant de la base fondamentale disponible $S(t)$. Un nombre fini d'étapes conduit à une nouvelle base fondamentale, dont l'ensemble d'optimalité contient un certain segment et a pour borne gauche $\bar{t}$, ou à l'établissement d'un certain ensemble (connexe) à borne gauche $\bar{t}$ sur lequel le problème n'a pas de solutions.

Dans le cas 3), on procède conformément aux recommandations du paragraphe 3.3, en transformant le vecteur des contraintes B de sorte que dans l'intervalle considéré la base fondamentale $S(t)$ soit une base. On voit aisément que cette transformation peut être réalisée en ajoutant à B une certaine combinaison linéaire des vecteurs de la base fondamentale, le résultat devant être en outre multiplié par (-1) dans le cas $\bar{t} = t^*$. Ensuite, après quelques itérations de la méthode primale il devient clair que les conditions du problème dual à (3.13)-(3.15) sont incompatibles, ou l'on construit une pseu-

dobase du problème (3.13)-(3.15). Dans les deux cas, les conditions indiquées sont remplies dans un certain intervalle de variation du paramètre t à borne gauche $\bar{t}$. Après l'obtention de la pseudobase, on rétablit le vecteur des contraintes du problème, et au terme de quelques étapes de la méthode duale du simplexe on construit une base fondamentale dont l'ensemble d'optimalité a une borne gauche $\bar{t}$, ou l'on découvre l'incompatibilité des conditions du problème dans un certain intervalle à borne gauche $\bar{t}$.

Lors de la discussion du problème (3.13)-(3.15), il suffit de fixer à chaque itération les paramètres $x_{ij}(t)$ ou les éléments $e_{ij}(t)$ du tableau principal pour une certaine valeur $t = t'$. Le choix du point t' n'est restreint que par les vecteurs des conditions qui constituent une base fondamentale pour un certain ensemble de l'axe des t, ces vecteurs devant être linéairement indépendants avec $t = t'$. En utilisant ces paramètres fixés, il est facile d'obtenir pour $x_{i0}(t)$ et $\Delta_j(t)$ les expressions nécessaires à l'exploration de la base fondamentale disponible. En effet, supposons pour fixer les idées que la résolution du problème s'obtienne à l'aide du deuxième algorithme (méthode de la matrice inverse). Soient $e_{ij} = e_{ij}(t')$, $i = 1, 2, \ldots, m$, $j = 1, 2, \ldots, m$, les éléments du tableau principal qui correspond au système de vecteurs $A_1(t')$, $A_{s_1}, \ldots, A_{s_m}$. Si x''_{i1}, $i = 1, 2, \ldots, m$, sont les coefficients de décomposition du vecteur A''_1 par rapport à la base considérée, et si

$$x''_{m+1,1} = \sum_{\alpha=1}^{m} e_{m+1,\alpha} a''_{\alpha 1}, \quad x_{1j} = \sum_{\alpha=1}^{m} e_{1\alpha} a_{\alpha j},$$

on a

$$x_{i0}(t) = e_{i0}(t) = \begin{cases} e_{i0} - \dfrac{e_{10}}{1+(t-t')x''_{11}}(t-t')x''_{i1}, & i = 2, 3, \ldots, m, \\ \dfrac{e_{10}}{1+(t-t')x''_{11}}, & i = 1; \end{cases}$$

$$\Delta_j(t) = \sum_{\alpha=1}^{m} a_{\alpha j} e_{m+1,\alpha}(t) = \Delta_j - \frac{x_{1j}}{1+(t-t')x''_{11}}(t-t')x''_{m+1,1}, \qquad j \neq 1.$$

Les formules ci-dessus se déduisent des relations de récurrence fondamentales de la programmation linéaire appliquées au cas où le vecteur $A_1(t')$ de la base fondamentale est remplacé par le vecteur $A_1(t) = A_1(t') + (t - t')A''_1$. Si la nécessité se présente, des formules analogues permettent de calculer $x_{ij}(t)$.

Lors de la discussion du problème (3.13)-(3.15), il est utile de se servir de la proposition suivante.

Théorème 3.3. *L'ensemble des valeurs du paramètre t pour lesquelles les conditions* (3.14)-(3.15) *sont compatibles est connexe.*

L'ensemble des valeurs du paramètre t pour lesquelles les conditions du problème dual par rapport à (3.13)-(3.15) *sont incompatibles est également connexe.*

La démonstration du théorème 3.3 fait l'objet de l'exercice 12.

L'exploration du problème paramétrique à droite de $\bar{t}$ impose, comme nous l'avons vu, la construction d'un nouvel ensemble d'optimalité, ou bien nous conduit aux conditions du cas 2) (incompatibilité des conditions du problème initial) ou du cas 3) (incompatibilité des conditions du problème dual). Dans le cas 2), la proposition du théorème 3.3 permet de conclure que les conditions (3.14)-(3.15) sont incompatibles pour tout point à droite de $\bar{t}$.

La procédure décrite [les cas 2) et 3) se discutent de façon analogue au cas 1) déjà examiné] permet d'étudier complètement le problème et conduit à la formation en un nombre fini d'itérations de la fonction

$$f(t) = \max_X \left(\sum_{j=1}^{n} c_j x_j\right)$$

[le maximum est pris par rapport aux points X qui vérifient les conditions (3.14)-(3.15)]. En appliquant l'algorithme donné et le théorème 3.3, on peut démontrer la proposition suivante.

Théorème 3.4. *Si le problème* (3.13)-(3.15) *a une solution réalisable pour plus d'une valeur de t, alors*

a) le domaine de définition E de la fonction $f(t)$ est un ensemble connexe ou bien est constitué de deux composantes connexes;

b) l'ensemble E est coupé par un nombre fini de points t_i (points critiques) en ensembles connexes E_i tels que sur tout E_i, $f(t)$ est une fonction homographique continue.

A la différence des cas examinés dans les paragraphes précédents $f(t)$ ne doit pas nécessairement être une fonction continue dans son domaine de définition. Elle peut subir une discontinuité aux points frontières des ensembles E_i, et de plus, les limites à gauche ou à droite aux points de discontinuité peuvent être aussi bien infinies que finies. L'algorithme donné ci-dessus permet d'analyser le problème (3.13)-(3.16) pour toutes les valeurs du paramètre t, à l'exception des points de discontinuité de la fonction $f(t)$. L'étude du problème pour ces valeurs fixées de t rend nécessaire un examen supplémentaire.

A titre de conclusion, indiquons que la mise en œuvre des considérations exposées dans ce paragraphe et les paragraphes précédents permet d'ébaucher sans peine un algorithme pour l'étude du problème (3.1)-(3.3) dans lequel la dépendance du paramètre n'intervient que dans les éléments d'une des lignes de la matrice des conditions. La composition du schéma général de cet algorithme est laissée au lecteur (cf. exercice 16). Parmi les ouvrages consacrés au problème (3.13)-(3.15), signalons surtout [44].

3.6. L'étude des problèmes (3.4)-(3.6) et (3.13)-(3.15) exposée aux paragraphes 3.2 à 3.5 peut servir de base à la construction d'un algorithme de programmation paramétrique appliqué au problème (3.1)-(3.3). Esquissons ce schéma dans ses grandes lignes: comme d'habitude, il faut commencer la discussion par la résolution du problème pour une certaine valeur de $t = t_0$.

Supposons, pour fixer les idées, qu'à une certaine itération de la procédure de résolution on ait obtenu le programme optimal du problème associé au système de vecteurs $A_{i_1}(t), A_{i_2}(t), \ldots, A_{i_m}(t)$. Décomposons le vecteur des contraintes et les vecteurs des conditions par rapport aux vecteurs de ce système (pour ceux des t pour lesquels il constitue une base fondamentale):

$$A_0(t) = B = \sum_{i=1}^{m} x_{i0}(t) A_{s_i}(t),$$

$$A_j(t) = \sum_{i=1}^{m} x_{ij}(t) A_{s_i}(t), \quad j = 1, 2, \ldots, n.$$

Les critères $\Delta_j(t)$ des vecteurs des conditions $A_j(t)$ par rapport à la base fondamentale choisie se calculent suivant les formules

$$\Delta_j(t) = \sum_{i=1}^{m} c_{s_i}(t) x_{ij}(t) - c_j(t), \quad j = 1, 2, \ldots, n.$$

Le système de vecteurs (base fondamentale) ainsi obtenu est une base de la solution du problème dans la marge de variation du paramètre t, qui vérifie les inégalités

$$x_{i0}(t) \geqslant 0, \quad i = 1, 2, \ldots, m, \tag{3.21}$$

$$\Delta_j(t) \geqslant 0, \quad j = 1, 2, \ldots, n, \tag{3.22}$$

on suppose de plus que le déterminant $\delta(t)$ du système de vecteurs considéré n'est pas nul.

En résolvant le système d'inégalités (3.21)-(3.22) compte tenu de la restriction $\delta(t) \neq 0$, on obtient l'ensemble d'optimalité de la base fondamentale considérée. Dans le cas général, les premiers membres des inégalités (3.21)-(3.22) se présentent comme des rapports entre certains polynomes et t. C'est pourquoi la construction de l'ensemble d'optimalité se ramène à la résolution d'une série d'équations algébriques. Il est clair que dans le cas général, l'ensemble d'optimalité est constitué de plusieurs composantes. Pour une étude plus poussée, il convient de choisir l'un des points frontières (le point t_1) de l'ensemble d'optimalité et d'aborder l'exploration du problème dans celui des intervalles suffisamment petits, adjacents à t_1, qui se trouve dans la partie de l'axe des t non examinée encore. Supposons,

pour fixer les idées, que cet intervalle soit $(t_1, t_1 + \varepsilon)$. Cherchons la solution du problème (3.1)-(3.3) pour $t = t_1 + \varepsilon$, où ε est un nombre positif suffisamment petit. On prend pour base fondamentale celle qui est une base optimale pour $t = t_1 - \varepsilon$. Il faut utiliser de plus les recommandations données dans les paragraphes 3.3 et 3.5. Conformément à ces recommandations, on obtient en quelques itérations de la méthode primale et (ou) duale du simplexe une base optimale pour l'intervalle $(t_1, t_1 + \varepsilon)$, ou bien suivant les propriétés d'un certain système de vecteurs des conditions on révélera que dans ce même intervalle le problème est irrésoluble.

Pour la nouvelle base fondamentale on procède d'une façon analogue. Il en résulte un ensemble des valeurs du paramètre tel qu'il garde les propriétés de la base fondamentale qui ont lieu pour l'intervalle $(t_1, t_1 + \varepsilon)$ (conditions d'optimalité ou de solution irréalisable).

On choisit ensuite l'un des points frontières de l'ensemble des valeurs étudiées du paramètre t, dont un voisinage aussi petit que l'on veut contient des points qui n'appartiennent pas à cet ensemble. En partant de ce point, on définit le nouvel ensemble d'optimalité ou de solution irréalisable.

La procédure s'achève après un nombre fini d'étapes, puisque le nombre total des bases du problème qui peuvent être composées de m vecteurs des conditions est fini.

Finalement, le problème paramétrique sera exploré pour toute la marge imposée de variation du paramètre t, sauf, peut-être, pour un nombre fini de points. Dans certains cas une discussion spéciale du problème est nécessaire pour ces valeurs particulières du paramètre.

Notons que les valeurs particulières du paramètre sont caractérisées par la propriété suivante : avec $t = t^*$, les systèmes de vecteurs des conditions qui engendrent les ensembles d'optimalité ou de solutions irréalisables et qui adhèrent des deux côtés au point singulier t^* sont linéairement dépendants.

Le schéma de l'algorithme énoncé peut être transposé à une classe suffisamment large de relations non linéaires des conditions du problème et du paramètre t. Ceci concerne, par exemple, le cas où les conditions du problème constituent des fonctions analytiques du paramètre.

L'hypothèse d'une dépendance analytique entre les données du problème et le paramètre t assure qu'avec un $\varepsilon > 0$ assez petit une base optimale du problème pour $t = t_1 + \varepsilon$ reste optimale pour tout $t \in (t_1, t_1 + \varepsilon]$, et les conditions d'irrésolubilité du problème avec $t = t_1 + \varepsilon$ se conservent pour tout $t \in (t_1, t_1 + \varepsilon]$. Cette conclusion résulte d'une propriété connue des fonctions analytiques : un point où une fonction non identiquement nulle est analytique ne peut être une limite pour les zéros de cette fonction.

3.7. Voici un exemple de problème général de programmation paramétrique linéaire emprunté à [67] *).

Maximiser pour toutes les valeurs de t $(-\infty < t < \infty)$ la forme linéaire

$$L(t) = -(30 + 6t)\, x_1 - (50 + 7t)\, x_2$$

sous les conditions

$$(14 + 2t)\, x_1 + (4 + t)\, x_2 + x_3 = 14 + t,$$
$$(150 + 3t)\, x_1 + (200 + 4t)\, x_2 - x_4 = 200 + 2t,$$
$$x_j \geqslant 0,\ j = 1, 2, 3, 4.$$

Avec $t = 0$, le programme optimal du problème est égal à $X(0) = \left(\frac{10}{11}, \frac{7}{22}, 0, 0\right)$. La base de la solution se compose des vecteurs A_1 et A_2. Décomposons le vecteur des contraintes $A_0(t)$ et les vecteurs des conditions $A_j(t)$ suivant les vecteurs $A_1(t)$ et $A_2(t)$ et formons les expressions des critères $\Delta_j(t)$ des vecteurs des conditions intervenant dans la base adoptée. Portons tous les paramètres du problème dans les tableaux de la méthode primale du simplexe (tableau 3.3).

La marge de variation de t dans laquelle les vecteurs A_1 et A_2 constituent une base de la solution vérifie le système d'inégalités (4.5)-(4.6). Dans notre cas, nous avons:

$$x_1(t) = \frac{\xi_1(t)}{\xi(t)} = \frac{2t^2 + 48t + 2000}{5t^2 + 294t + 2200} \geqslant 0,$$
$$x_2(t) = \frac{\xi_2(t)}{\xi(t)} = \frac{t^2 + 236t + 700}{5t^2 + 294t + 2200} \geqslant 0,$$
$$\Delta_3(t) = \frac{\delta_3(t)}{\xi(t)} = \frac{-3t^2 - 120t + 1500}{5t^2 + 294t + 2200} \geqslant 0,$$
$$\Delta_4(t) = \frac{\delta_4(t)}{\xi(t)} = \frac{8t^2 + 144t + 580}{5t^2 + 294t + 2200} \geqslant 0.$$

Dans l'expression relative à $x_1(t)$, le numérateur $\xi_1(t)$ est positif pour toute valeur réelle de t. La condition $x_1(t) \geqslant 0$ est remplie si on vérifie l'inégalité

$$\xi(t) = 5t^2 + 294t + 2200 > 0,$$

ou

$$(t + 8{,}8)(t + 50) > 0.$$

Cette dernière inégalité a lieu si

$$t < -50 \text{ ou } t > -8{,}8.$$

La condition $x_2(t) \geqslant 0$ est compatible avec la condition $x_1(t) \geqslant 0$ pour $\xi_2(t) \geqslant 0$. La valeur approchée de cette relation s'écrit

$$(t + 3)(t + 233) \geqslant 0.$$

On en tire

$$t \leqslant -233 \quad \text{ou} \quad t \geqslant -3.$$

---

*) Dans [67], lors du découpage de l'axe des t en intervalles tels que la base de la solution reste invariable on utilise à tort la seule condition $\Delta_j \geqslant 0$. C'est pourquoi, dans certaines marges de variation de t, les composantes de la solution donnée par [67] s'avèrent négatives.

Tableau 3.3

N	C_X	B_X	A_0	A_1	A_2	A_3	A_4
$\leftarrow 1$	$-(30+6t)$	A_1	$\dfrac{2t^2+48t+2000}{5t^2+294t+2200}$	1		$\dfrac{200+4t}{5t^2+294t+2200}$	$\dfrac{4+t}{5t^2+294t+2200}$
2	$-(50+7t)$	A_2	$\dfrac{t^2+236t+700}{5t^2+294t+2200}$		1	$-\dfrac{150+3t}{5t^2+294t+2200}$	$-\dfrac{14+2t}{5t^2+294+2200}$
Δ	—	—	$-\dfrac{19t^3+2050t^2+30\,140t+95\,000}{5t^2+294t+2200}$			$\dfrac{-3t^2-120t+1500}{5t^2+294t+2200}$	$\dfrac{8t^2+144t+580}{5t^2+294t+2200}$

D'une façon analogue, on obtient que la condition $\Delta_3(t) \geqslant 0$ est remplie pour

$$-50 \leqslant t \leqslant 10,$$

et que la condition $\Delta_4(t) \geqslant 0$ est vérifiée avec $t \leqslant -11{,}9$ ou $t \geqslant -6{,}1$.

L'ensemble des valeurs de t pour lesquelles les vecteurs A_1, A_2 constituent une base de la solution est ainsi déterminé par les inégalités

$$-3 \leqslant t \leqslant 10. \tag{3.23}$$

Afin de poursuivre la discussion, calculons la base optimale du problème pour un point quelconque $t_1 < -3$ ou $t_1 > 10$ voisin des extrémités du segment (3.23). Avec $t > 10$, la condition $\Delta_3 \geqslant 0$ est compromise. Par suite, le vecteur A_3 doit être introduit dans la base. Résolvons le problème pour $t_1 = 11 > 10$. La base du programme optimal est constituée par les vecteurs $A_2(t)$ et $A_3(t)$. Décomposons le vecteur des contraintes et les vecteurs des conditions par rapport aux vecteurs de cette base. Nous avons alors le tableau 3.4.

La marge de variation de t pour laquelle les vecteurs $A_2(t)$ et $A_3(t)$ constituent une base optimale est déterminée par le système d'inégalités

$$\frac{2t^2+48t+2000}{200+4t} \geqslant 0, \qquad \frac{3t^2+120t-1500}{200+4t} \geqslant 0,$$

$$\frac{200+2t}{200+4t} \geqslant 0, \qquad \frac{50+7t}{200+4t} \geqslant 0.$$

Le trinôme $2t^2 + 48t + 2000 \geqslant 0$ quelles que soient les valeurs de t. Aussi ce dernier système d'inégalités peut être mis sous la forme

$$200 + 4t > 0,$$
$$200 + 2t \geqslant 0,$$
$$(t + 50)(t - 10) \geqslant 0,$$
$$50 + 7t \geqslant 0.$$

Ce système d'inégalités se vérifie avec $t \geqslant 10$. Les vecteurs $A_2(t)$, $A_3(t)$ constituent ainsi une base optimale pour tout $t \geqslant 10$.

En nous appuyant sur les vecteurs $A_2(t)$, $A_3(t)$, cherchons s'il existe un domaine de modification de t tel que la forme linéaire du problème ne soit pas bornée sur l'ensemble de ses programmes. A cette fin trouvons la marge de variation du paramètre t dont les points satisfont aux conditions de solution irréalisable du problème pour les vecteurs $A_2(t)$, $A_3(t)$. Le système d'inégalités

$$\Delta_1(t) < 0, \; x_{11}(t) \leqslant 0, \; x_{21}(t) \leqslant 0$$

est contradictoire, du fait que $x_{21}(t) = \frac{3}{4} > 0$. Le système

$$\Delta_4(t) < 0, \; x_{14}(t) \leqslant 0, \; x_{24}(t) \leqslant 0$$

est compatible avec $-50 < t \leqslant -\frac{50}{7}$. Dans cette marge de variation de t, le problème n'a donc pas de solution du fait que sa forme linéaire est infinie.

Explorons le problème pour $t < -3$. Lorsque $t < -3$, pour $t = -4$ par exemple, les vecteurs des conditions $A_1(t)$, $A_3(t)$ constituent une base optimale. Calculons la marge de variation du paramètre t telle que les vecteurs $A_1(t)$, $A_3(t)$ constituent une base de la solution. Dressons le tableau 3.5.

Il est facile de voir que les vecteurs $A_1(t)$ et $A_3(t)$ restent base optimale du problème pour toutes les valeurs de t qui satisfont aux inégalités

$$-5 \leqslant t \leqslant -3 \quad \text{ou} \quad -\infty < t < -233.$$

En outre, pour $-\frac{50}{7} \leqslant t \leqslant -6{,}1$, la forme linéaire du problème n'est pas bornée sur l'ensemble de ses programmes ($\Delta_2(t) < 0$, $x_{12}(t) \leqslant 0$, $x_{22}(t) \leqslant 0$).

Tableau 3.4

N	C_X	B_X	A_0	A_1	A_2	A_3	A_4
$\rightarrow 1$		A_3	$\frac{2t^2+48t+2000}{200+4t}$	$\frac{5t^2+294t+2200}{200+4t}$		1	$\frac{4+t}{100+4t}$
$\leftarrow 2$	$-(50+7t)$	A_2	$\frac{200+2t}{200+4t}$	$\frac{150+3t}{200+4t}$	1		$-\frac{1}{200+4t}$
Δ	—	—	$-\frac{14t^2+1500t+10\,000}{200+4t}$	$\frac{3t^2+120t-1500}{200+4t}$			$\frac{50+7t}{200+4t}$

Tableau 3.5

N	C_X	B_X	A_0	A_1	A_2	A_3	A_4
$\leftarrow 1$		A_3	$-\dfrac{t^2+236t+700}{150+3t}$		$-\dfrac{5t^2+294t+2200}{150+3t}$	1	$\dfrac{14+2t}{150+3t}$
$\rightarrow 2$	$-(30+6t)$	A_1	$\dfrac{200+2t}{150+3t}$	1	$\dfrac{200+4t}{150+3t}$		$-\dfrac{1}{150+3t}$
Δ	—	—	$-\dfrac{12t^2+1260t+6000}{150+3t}$		$-\dfrac{3t^2+120t-1500}{150+3t}$		$\dfrac{30+6t}{150+3t}$

Composons les tableaux 3.6 à 3.8 de la méthode primale du simplexe associés aux systèmes de vecteurs $(A_4(t), A_1(t))$, $(A_4(t), A_2(t))$ et $(A_4(t), A_3(t))$.

L'analyse des tableaux 3.6 à 3.8, analogue à celle qui a été effectuée pour les tableaux 3.3 à 3.5, permet d'énoncer les conclusions siuvantes compte tenu des résultats obtenus :

1. Le problème exploré est réalisable pour toutes les valeurs du paramètre t qui appartiennent aux deux demi-droites $t \leqslant t_1$ et $t \geqslant t_2$ où $t_1 = -118 - 2\sqrt{3\,306} \approx -233$, $t_2 = -9 + \frac{1}{2}\sqrt{34} \approx -6{,}1$. De plus:

a) pour $t < -233$ et $-5 \leqslant t \leqslant -3$, la base optimale se compose des vecteurs $A_1(t)$, $A_3(t)$, le programme optimal est égal à

$$X = \left(\frac{200+2t}{150+3t},\ 0,\ -\frac{t^2+236t+700}{150+3t},\ 0\right),$$

et la valeur maximale de la forme linéaire se calcule selon la formule

$$L^* = -\frac{4(t^2+105t+500)}{50+t};$$

b) pour $-6{,}1 < t \leqslant -5$, la base de la solution se compose des vecteurs $A_1(t)$, $A_4(t)$, le programme optimal est égal à

$$X = \left(\frac{14+t}{14+2t},\ 0,\ 0,\ -\frac{t^2+236t+700}{14+2t}\right),$$

et la valeur de la forme linéaire dans le cas d'un programme optimal s'écrit

$$L^* = -\frac{3(t^2+19t+70)}{7+t};$$

c) pour $-3 \leqslant t \leqslant 10$, la base optimale est constituée des vecteurs $A_1(t)$, $A_2(t)$, le programme optimal est égal à

$$X = \left(\frac{2t^2+48t+2000}{5t^2+294t+2200},\ \frac{t^2+236t+700}{5t^2+294t+2200},\ 0,\ 0\right),$$

et la valeur maximale de la forme linéaire est

$$L^* = -\frac{19t^3+2050t^2+30140t+9500}{5t^2+294t+2200},$$

d) pour la demi-droite $t \geqslant 10$, la base est constituée des vecteurs $A_2(t)$, $A_3(t)$, le programme optimal est

$$X = \left(0,\ \frac{100+t}{100+2t},\ \frac{t^2+24t+1000}{100+2t},\ 0\right),$$

et la valeur maximale de la forme linéaire

$$L^* = -\frac{7t^2+750t+5000}{100+2t}.$$

2. Dans l'intervalle de variation du paramètre $(-50;\ -6{,}1)$, la forme linéaire du problème n'est pas bornée sur l'ensemble de ses programmes. (Pour $-50 < t < -7$, le critère Δ_4 du vecteur des conditions $A_4(t)$ par rapport à la base $A_2(t)$, $A_3(t)$ est négatif, et en même temps $x_{i4} \leqslant 0$; pour $-7 \leqslant t < -6{,}1$, le critère Δ_2 du vecteur $A_2(t)$ par rapport à la base $A_1(t)$, $A_3(t)$ est négatif et de plus tout $x_{i2} < 0$.)

3. Dans la marge de variation du paramètre $-233 \leqslant t \leqslant -50$, le problème n'a pas de solution, ses conditions étant incompatibles. Pour n'importe

Tableau 3.6-3.8

N	C_X	B_X	A_0	A_1	A_2	A_3	A_4
$\rightarrow 1$		A_4	$-\frac{t^2+236t+700}{14+2t}$		$-\frac{5t^2+294t+2200}{14+2t}$	$\frac{150+3t}{14+2t}$	1
$\leftarrow 2$	$-(30+6t)$	A_1	$\frac{14+t}{14+2t}$	1	$\frac{4+t}{14+2t}$	$\frac{1}{14+2t}$	
Δ	—	—	$-\frac{6t^2+114t+420}{14+2t}$		$\frac{8t^2+144t+580}{14+2t}$	$-\frac{30+6t}{14+2t}$	
1		A_4	$\frac{2t^2+48t+2000}{4+t}$	$\frac{5t^2+294t+2200}{4+t}$		$\frac{200+4t}{4+t}$	1
$\rightleftarrows 2$	$-(50+7t)$	A_3	$\frac{14+t}{4+t}$	$\frac{14+2t}{4+t}$	1	$\frac{1}{4+t}$	
Δ	—	—	$-\frac{7t^2+148t+700}{4+t}$	$-\frac{8t^2+144t+580}{4+t}$		$-\frac{50+7t}{4+t}$	
1		A_4	$-(200+2t)$	$-(150+3t)$	$-(200+4t)$		1
$\rightarrow 2$		A_3	$14+t$	$14+2t$	$4+t$	1	
Δ	—	—		$30+6t$	$50+7t$		

laquelle des bases du problème, l'une des composantes au moins de la décomposition du vecteur des contraintes suivant les vecteurs de la base considérée est négative pour tout t appartenant au segment [—233, —50].

§ 4. Applications de la programmation paramétrique

4.1. Il existe pour plusieurs modèles particuliers de programmation linéaire des méthodes spéciales de discussion qui tiennent compte de la structure de leurs mátrices des conditions. Ces méthodes spéciales auxquelles on associe le problème de transport, le problème d'affectation, etc., simplifient nettement la procédure. Mais que l'on ajoute aux conditions de ces problèmes particuliers une condition au moins de type général, et ces méthodes spéciales simples perdent leurs avantages et ne peuvent plus être utilisées directement. On voudrait, naturellement, pouvoir appliquer les méthodes économiques de résolution du problème particulier correspondant à la discussion d'un problème plus complexe. Nous allons montrer dans ce qui suit que l'application des principes de programmation paramétrique permet quelquefois de ramener l'exploration d'un problème de programmation linéaire à un modèle particulier qui lui est apparenté [23].

Considérons le problème de programmation linéaire qui consiste à maximiser la forme linéaire

$$\sum_{j=1}^{n} c_j x_j \tag{4.1}$$

sous les conditions

$$\sum_{j=1}^{n} a_{ij} x_j = b_i, \quad i = 1, 2, \ldots, m; \tag{4.2}$$

$$x_j \geqslant 0, \; j = 1, 2, \ldots, n; \tag{4.3}$$

$$\sum_{j=1}^{n} d_j x_j \leqslant d. \tag{4.4}$$

Supposons que les conditions (4.2) aient une structure simple, de sorte qu'il est facile d'effectuer la discussion du problème (4.1)-(4.3) pour n'importe quelles valeurs des coefficients c_j à l'aide de la méthode primale du simplexe spécialement adaptée à ce modèle. Le problème (4.1)-(4.3) diffère du problème simple (4.1)-(4.3) par une seule condition de type général. Montrons comment utiliser, pour résoudre le problème (4.1)-(4.4), la méthode élaborée pour l'étude des modèles du type (4.1)-(4.3).

Introduisons dans la discussion le problème A_λ où l'on demande de maximiser la forme linéaire

$$\sum_{j=1}^{n} (c_j - \lambda d_j) x_j \tag{4.5}$$

sous les conditions (4.2), (4.3).

Supposons pour simplifier l'analyse que l'ensemble des points X défini par les conditions (4.2), (4.3) soit borné et constitue, par conséquent, un polyèdre convexe.

L'algorithme de programmation paramétrique exposé au § 1 ramène la discussion de la famille de problèmes A_λ ($\lambda \geqslant 0$) au découpage de l'axe des λ positif en une série de segments $[0, \lambda_1]$, $[\lambda_1, \lambda_2]$, . . ., $[\lambda_{s-1}, \lambda_s]$ et une demi-droite $\lambda \geqslant \lambda_s$ tels que pour tout $\lambda \in [\lambda_{k-1}, \lambda_k]$ le programme X_k ($k = 1, 2, \ldots, s$) est une solution du problème A_λ et le programme X_{s+1} *) une solution du problème A_λ quel que soit $\lambda \geqslant \lambda_s$. Nous avons vu au § 1 que le passage du programme X_k au programme X_{k+1} s'effectue en une ou plusieurs itérations de la méthode primale du simplexe adaptée au problème défini par les conditions (4.2), (4.3).

Nous allons énoncer et démontrer quelques propriétés des programmes X_k, qui permettent d'utiliser ces derniers pour la construction de la solution cherchée du problème (4.1)-(4.4).

L e m m e 4.1. *Lorsqu'on passe du programme X_k au programme X_{k+1}, la valeur de la forme linéaire*

$$\sum_{j=1}^{n} d_j x_j$$

décroît, c'est-à-dire

$$L_k = \sum_{j=1}^{n} d_j x_j^{(k)} > L_{k+1} = \sum_{j=1}^{n} d_j x_j^{(k+1)}. \tag{4.6}$$

D é m o n s t r a t i o n. Le programme X_{k+1} est une solution du problème A_λ avec $\lambda_k < \lambda < \lambda_{k+1}$. Le programme X_k ne constitue pas une solution du problème dans cette marge de variation du paramètre. C'est pourquoi

$$\begin{aligned}\sum_{j=1}^{n} (c_j - \lambda d_j)\, x_j^{(k)} &= \sum_{j=1}^{n} (c_j - \lambda_k d_j)\, x_j^{(k)} - \\ &- (\lambda - \lambda_k) \sum_{j=1}^{n} d_j x_j^{(k)} < \sum_{j=1}^{n} (c_j - \lambda d_j)\, x_j^{(k+1)} = \\ &= \sum_{j=1}^{n} (c_j - \lambda_k d_j)\, x_j^{(k+1)} - (\lambda - \lambda_k) \sum_{j=1}^{n} d_j x_j^{(k+1)}. \end{aligned} \tag{4.7}$$

*) Dans un cas particulier, l'ensemble des segments peut être vide. Dans ce cas, $\lambda_s = 0$.

Les programmes X_k et X_{k+1} sont les solutions du problème A_{λ_k}. Par suite,

$$\sum_{j=1}^{n} (c_j - \lambda_k d_j)\, x_j^{(k)} = \sum_{j=1}^{n} (c_j - \lambda_k d_j)\, x_j^{(k+1)}. \tag{4.8}$$

L'inégalité (4.7) avec l'égalité (4.8) entraînent la relation (4.6).

Lemme 4.2. *Si*

$$L_1 = \sum_{j=1}^{n} d_j x_j^{(1)} \leqslant d, \tag{4.9}$$

X_1 *est alors une solution du problème* (4.1)-(4.4). *Si avec un certain* $1 \leqslant k \leqslant s$

$$L_k = \sum_{j=1}^{n} d_j x_j^{(k)} > d \geqslant L_{k+1} = \sum_{j=1}^{n} d_j x_j^{(k+1)}, \tag{4.10}$$

le problème (4.1)-(4.4) *a pour solution le programme*

$$X^* = tX_k + (1-t)\, X_{k+1}, \tag{4.11}$$

où

$$t = \frac{d - L_{k+1}}{L_k - L_{k+1}}. \tag{4.12}$$

Démonstration. Le programme X_1 est une solution du problème (4.1)-(4.3), qui coïncide avec le problème A_λ pour $\lambda = 0$. Il est évident que si les conditions (4.9) sont vérifiées, X_1 est également une solution du problème (4.1)-(4.4).

Supposons maintenant que l'inégalité (4.10) ait lieu pour un certain $1 \leqslant k \leqslant s$. En vertu de (4.10),

$$0 \leqslant d - L_{k+1} < L_k - L_{k+1}.$$

Donc, $0 \leqslant t \leqslant 1$, et par conséquent, X^* étant une combinaison convexe des programmes X_k et X_{k+1} du problème (4.1)-(4.3), il satisfait aux conditions (4.2), (4.3). La substitution directe montre que

$$\sum_{j=1}^{n} d_j x_j^* = d. \tag{4.13}$$

Donc, X^* est un programme du problème (4.1)-(4.4). Le vecteur X^* est en outre une solution du problème A_{λ_k}, puisque X_k et X_{k+1} sont des programmes optimaux de ce problème. D'après le critère d'optimalité, pour les programmes du problème A_{λ_k} il existe des nombres $\mu_1, \mu_2, \ldots, \mu_m$ (critères des conditions du problème A_{λ_k})

tels que

$$\begin{aligned} c_j - \lambda_k d_j &\leqslant \sum_{i=1}^{m} a_{ij}\mu_i, \quad j=1, 2, \ldots, n; \\ c_j - \lambda_k d_j &= \sum_{i=1}^{m} a_{ij}\mu_i \text{ si } x_j^* > 0. \end{aligned} \tag{4.14}$$

Les conditions (4.14) et l'égalité (4.13) traduisent le fait que les nombres $\mu_1, \mu_2, \ldots, \mu_m, \lambda_k > 0$ sont les critères de sélection des conditions du problème (4.1)-(4.4). Par conséquent, le programme X^* est une solution de ce problème, ce qu'il fallait démontrer.

Lemme 4.3. *Si*

$$L_{s+1} = \sum_{j=1}^{n} d_j x_j^{(s+1)} > d,$$

les conditions (4.2)-(4.4) *sont contradictoires.*

Démonstration. Soit X un vecteur arbitraire qui vérifie les conditions (4.2), (4.3). Conformément aux notations adoptées, le vecteur X_{s+1} est une solution du problème A_λ avec $\lambda \geqslant \lambda_s$. Par suite,

$$\sum_{j=1}^{n} (c_j - \lambda d_j)\, x_j \leqslant \sum_{j=1}^{n} (c_j - \lambda d_j)\, x_j^{(s+1)} \text{ pour } \lambda \geqslant \lambda_s. \tag{4.15}$$

Divisons par λ les deux membres de l'inégalité (4.15) et passons à la limite pour $\lambda \to \infty$. Il vient

$$\sum_{j=1}^{n} d_j x_j \geqslant \sum_{j=1}^{n} d_j x_j^{(s+1)} = L_{s+1}.$$

Avec $L_{s+1} > d$, cette dernière inégalité conduit à la relation

$$\sum_{j=1}^{n} d_j x_j > d,$$

ce qui réduit à l'absurde la condition (4.4). Le lemme est ainsi démontré.

4.2. Les lemmes démontrés ci-dessus permettent de justifier la procédure de résolution du problème (4.1)-(4.4) fondée sur l'algorithme de programmation linéaire du § 1.

Cette procédure débute par la détermination du programme optimal X_0 du problème (4.1)-(4.3). Si le programme X_0 vérifie l'inégalité (4.4) d'après le lemme 4.2, il constitue une solution du problème (4.1)-(4.4). S'il n'en est pas ainsi, après avoir adopté X_0 comme programme initial, on procède à la discussion du problème paramétrique A_λ pour $\lambda \geqslant 0$. A cette fin on utilise l'algorithme du § 1. On sait que la procédure imposée par l'algorithme du § 1 se compose d'itérations de la méthode primale du simplexe réalisées

successivement en se guidant sur une règle spéciale de choix du vecteur A_k à inclure dans la base. Dans le cas donné, cette règle s'écrit

$$\frac{\Delta_{k1}^{(t)}}{\Delta_{k2}^{(t)}} = \min_{\Delta_{j2}^{(t)}>0} \frac{\Delta_{j1}^{(t)}}{\Delta_{j2}^{(t)}}, \quad \Delta_{k2}^{(t)} > 0. \tag{4.16}$$

Ici

$$\Delta_{j1}^{(t)} = \sum_{s=1}^{m} c_{i_s} x_{sj}^{(t)} - c_j; \quad \Delta_{j2}^{(t)} = \sum_{s=1}^{m} d_{i_s} x_{sj}^{(t)} - d_j,$$

$$y \parallel 1, 2, \ldots, n;$$

$x_{sj}^{(t)}$ sont les coefficients de décomposition du vecteur A_j par rapport aux vecteurs de la base $A_{i_1}, A_{i_2}, \ldots, A_{i_m}$ du programme X_t obtenu à la t-ième itération. Pour chaque programme X_t, on calcule

$$L_t = \sum_{j=1}^{n} d_j x_j^{(t)}.$$

La procédure s'achève par la construction d'un programme X_l qui vérifie l'une des deux conditions

$$1^\circ \; L_{l-1} > d, \quad L_l \leqslant d;$$

$$2^\circ \; L_l > d, \quad \Delta_{j2}^{(l)} \leqslant 0 \quad \text{pour tout } j.$$

Dans le cas 1°, le programme

$$X^* = \alpha X_{l-1} + (1 - \alpha) X_l \tag{4.17}$$

avec

$$\alpha = \frac{d - L_l}{L_{l-1} - L_l} \tag{4.18}$$

est une solution du problème (4.1)-(4.4) (cf. lemme 4.2). Si par contre on est dans le cas 2°, d'après le lemme 4.3, le problème (4.1)-(4.4) n'a pas de solution, les conditions (4.2)-(4.4) étant incompatibles.

Ainsi, l'application de la méthode primale du simplexe au problème de programmation linéaire défini par les conditions (4.2), (4.3) permet, en un nombre fini d'itérations, d'obtenir la solution du problème (4.1)-(4.4) ou d'établir que ses conditions sont incompatibles.

Dans le problème (4.1)-(4.4), la contrainte supplémentaire (4.4) avait la forme d'une inégalité. L'algorithme exposé ci-dessus sera très peu modifié si l'on remplace l'inégalité (4.4) par l'égalité

$$\sum_{j=1}^{n} d_j x_j = d. \tag{4.19}$$

Si le programme optimal X_0 du problème (4.1)-(4.3) vérifie la condition

$$\sum_{j=1}^{n} d_j x_j^{(0)} \geqslant d,$$

la procédure de résolution s'effectue d'après les règles déjà décrites. Mais si

$$\sum_{j=1}^{n} d_j x_j^{(0)} < d,$$

il faut procéder à la discussion des problèmes A_λ avec $\lambda \leqslant 0$. C'est pourquoi le choix du vecteur à inclure dans la base à la t-ième itération est déterminé non par la relation (4.16), mais par la condition

$$\frac{\Delta_{k1}^{(t)}}{\Delta_{k2}^{(t)}} = \max_{\Delta_{j2}^{(t)}<0} \frac{\Delta_{j1}^{(t)}}{\Delta_{j2}^{(t)}}, \quad \Delta_{k_2}^{(t)} < 0. \tag{4.20}$$

La règle d'achèvement de la procédure est elle aussi modifiée de façon correspondante: la l-ième itération est la dernière si

$$L_{l-1} < d, \quad L_l \geqslant d \quad \text{(cas 1°)}$$

ou si

$$L_l < d, \quad \Delta_{j_2}^{(l)} \geqslant 0 \quad \text{pour tout } j \quad \text{(cas 2°)}.$$

Dans le cas 1°, la solution X^* du problème (4.1)-(4.4) se calcule d'après les formules (4.17), (4.18). Dans le cas 2°, les conditions (4.2), (4.3), (4.19) sont incompatibles.

4.3. L'algorithme exposé peut être utilisé pour résoudre n'importe quel problème spécial de programmation linéaire rendu plus complexe par une contrainte de type général. Appliquons-le à la résolution d'un problème de transport classique à une condition supplémentaire de type général.

Supposons qu'il faille minimiser la forme linéaire

$$\sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} \tag{4.21}$$

sous les conditions du type « transport »;

$$\sum_{j=1}^{n} x_{ij} = a_i, \quad i = 1, 2, \ldots, m; \tag{4.22}$$

$$\sum_{i=1}^{m} x_{ij} = b_j, \quad j = 1, 2, \ldots, n; \tag{4.23}$$

$$x_{ij} \geqslant 0, \quad i = 1, 2, \ldots, m; \quad j = 1, 2, \ldots, n, \tag{4.24}$$

et une contrainte supplémentaire de type général

$$\sum_{i=1}^{m} \sum_{j=1}^{n} d_{ij} x_{ij} \leqslant d. \tag{4.25}$$

Pour résoudre le problème de transport (4.21)-(4.24), utilisons la méthode des potentiels (cf. par exemple, [87], § 1, 2, chapitre 14). La procédure de résolution du problème (4.21)-(4.24) s'achève par le calcul du programme optimal X_0 et de la matrice $\Delta_1^{(0)}$ aux éléments

$$\Delta_{ij1}^{(0)} = c_{ij} - (v_j - u_i) \geqslant 0,$$

où v_j, u_i sont les potentiels du problème (4.21)-(4.24).

S'il s'avère que X_0 vérifie la condition (4.25), ce programme est une solution du problème (4.21)-(4.25).

Dans le cas contraire, on poursuit le calcul des itérations de la méthode des potentiels. La différence entre ces itérations et les étapes usuelles de la méthode des potentiels réside en ce qu'à une itération quelconque (de numéro t) au lieu d'un seul système de potentiels on calcule deux systèmes:

$$u_i^{(t)},\ v_j^{(t)} \quad \text{et} \quad \bar{u}_i^{(t)},\ \bar{v}_j^{(t)}$$

définis par les matrices

$$C = \| c_{ij} \|_{m \cdot n} \quad \text{et} \quad D = \| d_{ij} \|_{m \cdot n}$$

respectivement.

Les indices i_0, j_0 du transport à inclure dans le nombre des transports de base à la t-ième itération sont choisis d'après la condition

$$-\frac{\Delta_{i_0 j_0 1}^{(t)}}{\Delta_{i_0 j_0 2}^{(t)}} = \min_{\Delta_{ij2}^{(t)} < 0} \left(-\frac{\Delta_{ij1}^{(t)}}{\Delta_{ij2}^{(t)}} \right), \quad \Delta_{i_0 j_0 2}^{(t)} < 0, \tag{4.26}$$

où

$$\Delta_{ij1}^{(t)} = c_{ij} - (v_j^{(t)} - u_i^{(t)}); \quad \Delta_{ij2}^{(t)} = d_{ij} - (\bar{v}_j^{(t)} - \bar{u}_i^{(t)}).$$

Posons

$$L_t = \sum_{i=1}^{m} \sum_{j=1}^{n} d_{ij} x_{ij}^{(t)}.$$

La procédure prend fin à la l-ième itération si

$$1^\circ \ L_{l-1} > d, \quad L_l \leqslant d$$

ou si

$$2^\circ \ L_l > d, \quad \Delta_{ij2}^{(l)} \geqslant 0 \quad \text{pour tout } i \text{ et } j.$$

Dans le cas 1°, la solution X^* du problème (4.21)-(4.25) est définie par les relations (4.17), (4.18), dans lesquelles il faut entendre par X_l le programme de transport construit à la t-ième itération. Dans le cas 2° on enregistre l'incompatibilité des conditions (4.22)-(4.25).

L'algorithme considéré est un cas particulier de la méthode générale exposée aux paragraphes 4.1, 4.2. La différence entre les conditions (4.26) et (4.16) s'explique par le fait que (4.21)-(4.25) est un problème de minimisation, ce qui entraîne que la forme linéaire du problème correspondant A_λ est de la forme

$$\sum_{i=1}^{m} \sum_{j=1}^{n} (c_{ij} + \lambda d_{ij}) x_{ij}.$$

4.4. Il existe des modèles de programmation linéaire dont tous les vecteurs des conditions sauf un (A_{n+1}, par exemple) ont une structure très simple, de sorte que si l'on fixe la variable x_{n+1}, la discussion de tels modèles devient bien plus simple.

Considérons le problème qui consiste à maximiser la forme linéaire

$$\sum_{j=1}^{n+1} c_j x_j \tag{4.27}$$

sous les conditions

$$\sum_{j=1}^{n+1} A_j x_j = B; \tag{4.28}$$

$$x_j \geqslant 0, \quad j = 1, 2, \ldots, n+1. \tag{4.29}$$

Supposons qu'un type particulier de la matrice

$$A = (A_1, A_2, \ldots, A_n)$$

rende possible l'application d'un algorithme spécial, suffisamment économique, se basant sur la méthode duale du simplexe à la résolution d'un problème de programmation linéaire à matrice des conditions A.

Si l'on pose que $x_{n+1} = \mu$ est le paramètre et si l'on met les conditions (4.28) sous la forme équivalente

$$\sum_{j=1}^{n} A_j x_j = B - \mu A_{n+1}, \tag{4.30}$$

il devient alors possible d'utiliser l'algorithme de programmation paramétrique exposé au § 2.

Nous donnerons le schéma général de la procédure en supposant que les conditions (4.29), (4.30) sont compatibles pour tout $\mu \geqslant 0$ (cf. exercice 20, où l'on propose au lecteur d'éliminer la contrainte indiquée).

1. Cherchons la solution du problème (4.27)-(4.29), dans lequel x_{n+1} est posé d'avance égal à zéro. Si le critère de sélection du vecteur A_{n+1} par rapport à la base optimale obtenue est non négatif, le programme construit est une solution du problème (4.27)-(4.29). Dans le cas contraire, on passe au stade suivant de la procédure.

2. Chaque étape de ce stade est une itération de la méthode duale du simplexe appliquée au problème à matrice A, dans lequel le vecteur à éliminer de la base est déterminé à l'aide d'une règle spéciale donnée ci-dessous.

Soient $x_{i0}^{(1)}$ et $x_{i0}^{(2)}$ les coefficients de décomposition suivant la base courante des vecteurs B et A_{n+1} respectivement. Le numéro r de la case de la base astreinte à être renouvelée est choisi d'après la relation

$$\frac{x_{r0}^{(1)}}{x_{r0}^{(2)}} = \min_{x_{i0}^{(2)}>0} \frac{x_{i0}^{(1)}}{x_{i0}^{(2)}}, \quad x_{r0}^{(2)} > 0. \tag{4.31}$$

3. Affectons de l'indice k toutes les quantités qui correspondent à la k-ième itération. $\Delta_{n+1}^{(k)}$ notamment est le critère du vecteur des conditions A_{n+1} par rapport à la base de la k-ième itération.

La procédure s'achève à la l-ième itération si

$$1^\circ \quad \Delta_{n+1}^{(l-1)} < 0, \quad \Delta_{n+1}^{(l)} \geqslant 0$$

ou si

$$2^\circ \quad \Delta_{n+1}^{(l)} < 0, \quad x_{i0}^{(2,\,l)} \leqslant 0 \quad \text{pour tout } i.$$

Dans le premier cas, le vecteur X^* aux composantes

$$x_j^* = \begin{cases} x_{i0}^{(1,l)} - \mu^* x_{i0}^{(2,\,l)} & \text{si } j = s_i, \quad i = 1, 2, \ldots, m; \\ \mu^* & \text{si } j = n+1; \\ 0 & \text{dans les autres cas} \end{cases}$$

est une solution du problème (4.27)-(4.29). μ^* est ici la valeur du minimum (4.31) correspondant à la $(l-1)$-ième itération; $(s_1, s_2, \ldots, s_m)$ sont les numéros des vecteurs de la base de la l-ième itération.

Si ce sont les conditions du cas 2° qui sont remplies, on en tire la conclusion que la forme linéaire (4.27) est infinie sur le polyèdre convexe (4.28), (4.29). L'algorithme exposé est l'analogue dual de la procédure étudiée de près aux paragraphes 4.1, 4.2. Sa justification est laissée au lecteur (cf. exercice 19).

4.5. Voici encore une application de l'algorithme du § 1 à la résolution du problème général de programmation linéaire proposée par [54].

Soit un problème de programmation linéaire qui consiste à maximiser la forme linéaire

$$\sum_{j=1}^{n} c_j x_j \tag{4.32}$$

sous les conditions

$$\sum_{j=1}^{n} a_{ij} x_j = b_i, \quad i = 1, 2, \ldots, m, \tag{4.33}$$

$$x_j \geqslant 0. \tag{4.34}$$

Sans restreindre la généralité, on peut, considérer toutes les composantes b_i du vecteur B comme non négatives.

Supposons que nous connaissions un programme $(\delta_1, \delta_2, \ldots, \delta_m)$ du problème dual de (4.32)-(4.34) et que, de plus, toutes les composantes δ_i soient strictement positives. Il est commode d'écrire les conditions (4.33), (4.34) sous la forme équivalente:

$$\sum_{j=1}^{n} a_{ij}x_j + x_{n+i} = b_i, \quad i = 1, 2, \ldots, m; \tag{4.35}$$

$$\sum_{i=1}^{m} \delta_i x_{n+i} = 0; \tag{4.36}$$

$$x_j \geqslant 0, \quad j = 1, 2, \ldots, n + m. \tag{4.37}$$

Introduisons dans la discussion le problème paramétrique A_λ, $\lambda \geqslant -1$, à la forme linéaire

$$\sum_{j=1}^{n} c_j x_j - \lambda \sum_{i=1}^{m} \delta_i x_{n+i}, \tag{4.38}$$

qu'il faut maximiser et aux conditions (4.35), (4.37).

On vérifie aisément que les vecteurs unités $A_{n+1} = e_1$, $A_{n+2} = e_2, \ldots, A_{n+m} = e_m$ constituent une base optimale du problème A_λ avec $\lambda = -1$. D'après les résultats obtenus aux paragraphes 4.1, 4.2 la résolution du problème (4.32), (4.35)-(4.37) se ramène à l'exploration de la famille de problèmes A_λ, $\lambda \geqslant -1$, pour une base initiale coïncidant avec la matrice unité $(e_1, e_2, \ldots, e_m)$. C'est pourquoi il ne nous reste qu'à traduire l'algorithme du paragraphe 4.2 en termes du problème donné.

Désignons par

$$\Delta_{j1}^{(t)} \quad \text{et} \quad \Delta_{j2}^{(t)} \quad (j = 1, 2, \ldots, n + m)$$

les critères de sélection des vecteurs des conditions A_j par rapport à la base de la t-ième itération de la procédure, associés respectivement aux formes linéaires

$$L_1 = \sum_{j=1}^{n} c_j x_j \quad \text{et} \quad L_2 = \sum_{i=1}^{m} \delta_i x_{n+i}.$$

Chaque itération de l'algorithme est une étape de la méthode primale du simplexe appliquée au problème soumis aux conditions (4.35), (4.37). Le vecteur A_k à inclure dans la base à la t-ième itération est défini par la relation

$$\frac{\Delta_{k1}^{(t)}}{\Delta_{k2}^{(t)}} = \min_{\Delta_{j2}^{(t)} > 0} \frac{\Delta_{j1}^{(t)}}{\Delta_{j2}^{(t)}}, \quad \Delta_{k2}^{(t)} > 0. \tag{4.39}$$

La procédure prend fin à la l-ième itération si l'on vérifie l'une des deux conditions suivantes:

$$1^\circ \quad L_2(X_l) = 0; \quad 2^\circ \quad L_2(X_l) > 0 \quad \text{et} \quad \Delta_{j2}^{(l)} \leqslant 0$$
$$(j = 1, \ldots, n + m).$$

($X_l = (x_1^{(l)}, x_2^{(l)}, \ldots, x_{n+m}^{(l)})$ est ici le programme associé à la l-ième itération).

Dans le cas 1°, les n premières composantes du vecteur X_l constituent une solution du problème (4.32)-(4.34) (les m dernières composantes de ce vecteur sont évidemment nulles). Cette conclusion se déduit du lemme 4.2. D'après le lemme 4.3, le cas 2° n'est possible que lorsque les conditions (4.35)-(4.37) ou, ce qui revient au même, les relations (4.33), (4.34) sont incompatibles.

Dans l'hypothèse de non-dégénérescence du problème (4.32)-(4.34), la valeur de la forme linéaire $L_2(X)$, variable d'écart pondérée des conditions (4.33), subit une décroissance monotone. L'algorithme décrit peut être utilisé au lieu de la M-méthode bien connue. Pour de nombreuses formulations pratiques, l'hypothèse de l'existence d'un programme du problème dual aux composantes positives n'est pas restrictive. Il en est ainsi, par exemple, des problèmes dans lesquels $A_j \geqslant 0$, $A_j \neq 0$ pour $j = 1, 2, \ldots, n$.

4.6. Dans les paragraphes suivants nous allons montrer que la méthode de programmation paramétrique du § 1 permet de réaliser la discussion numérique complète du problème de programmation homographique qui consiste à maximiser la fonction homographique

$$R(X) = \frac{\sum_{j=1}^{n} c_j x_j + c}{\sum_{j=1}^{n} d_j x_j + d} = \frac{L_1(X)}{L_2(X)} \tag{4.40}$$

sous les conditions

$$\sum_{j=1}^{n} a_{ij} x_j = b_i, \quad i = 1, 2, \ldots, m; \tag{4.41}$$

$$x_j \geqslant 0, \quad j = 1, 2, \ldots, n. \tag{4.42}$$

Pour abréger l'exposé, par la suite, nous supposerons borné l'ensemble M des points X qui vérifient les conditions (4.41), (4.42).

Soit

$$L_\lambda(X) = L_1(X) + \lambda L_2(X). \tag{4.43}$$

Introduisons les fonctions

$$f_1(\lambda) = \max_{X \in M} L_\lambda(X) \tag{4.44}$$

et

$$f_2(\lambda) = \min_{X \in M} L_\lambda(X). \tag{4.45}$$

On sait (cf. paragraphe 1.6) que $f_1(\lambda)$ ($f_2(\lambda)$) est définie dans le cas considéré sur l'axe tout entier et constitue une fonction continue linéaire par morceaux, convexe vers le bas (vers le haut).

Désignons par $X_{\lambda 1}^{(1)}$ et $X_{\lambda 2}^{(1)}$ les solutions du problème de maximisation de la fonction $L_{\lambda_1}(X)$, sous les conditions (4.41), (4.42), pour des λ_1 disposés respectivement sur les segments $[\lambda, \lambda + \varepsilon]$ et $[\lambda - \varepsilon, \lambda]$, où ε est un nombre positif suffisamment petit. (Il est clair que ε doit être choisi de sorte que les intervalles $(\lambda, \lambda + \varepsilon)$ et $(\lambda - \varepsilon, \lambda)$ ne contiennent pas de points critiques de la fonction $f_1(\lambda)$.)

Introduisons de façon analogue, en partant du problème de minimisation de $L_\lambda(X)$ et sous les mêmes conditions

$$X_{\lambda 1}^{(2)} \quad \text{et} \quad X_{\lambda 2}^{(2)}.$$

Les vecteurs $X_{\lambda 1}^{(1)}$ et $X_{\lambda 2}^{(1)}$ ($X_{\lambda 1}^{(2)}$ et $X_{\lambda 2}^{(2)}$) ne diffèrent que si λ est un point critique de la fonction f_1 (f_2).

Soient $f'_{11}(\lambda)$ et $f'_{12}(\lambda)$ les dérivées à droite et à gauche de la fonction $f_1(\lambda)$; $f'_{21}(\lambda)$ et $f'_{22}(\lambda)$ les dérivées à droite et à gauche de la fonction $f_2(\lambda)$. De la définition des vecteurs $X_{\lambda j}^{(i)}$ découlent les égalités

$$\begin{aligned} f'_{11}(\lambda) &= L_2(X_{\lambda 1}^{(1)}), \quad f'_{12}(\lambda) = L_2(X_{\lambda 2}^{(1)}), \\ f'_{21}(\lambda) &= L_2(X_{\lambda 1}^{(2)}), \quad f'_{22}(\lambda) = L_2(X_{\lambda 2}^{(2)}). \end{aligned} \tag{4.46}$$

L'algorithme de résolution du problème (4.40)-(4.42) se fonde sur certaines propriétés des fonctions $f_1(\lambda)$ et $f_2(\lambda)$. Ces propriétés se déduisent de deux propositions auxiliaires.

Soit $q_{\max}(q_{\min})$ la valeur maximale (minimale) de $L_2(X)$ sous les conditions (4.41), (4.42).

Désignons par r la face supérieure de la fonction $R(X)$ avec $X \in M$ et par $r(q)$ ($q \neq 0$, $q_{\min} \leqslant q \leqslant q_{\max}$) la face supérieure de $R(X)$ avec $X \in M$, $L_2(X) = q$.

L e m m e 4.4. *Soit $q \neq 0$ un nombre arbitraire compris entre $q_{\min}$ et $q_{\max}$.*

Si $q > 0$ et si λ est déterminé par les conditions

$$f'_{12}(\lambda) \leqslant q \leqslant f'_{11}(\lambda), \tag{4.47}$$

il vient

$$r(q) = R(\alpha X_{\lambda 1}^{(1)} + (1 - \alpha) X_{\lambda 2}^{(1)}), \tag{4.48}$$

où le coefficient α se calcule d'après la relation

$$\alpha L_2(X_{\lambda 1}^{(1)}) + (1 - \alpha) L_2(X_{\lambda 2}^{(1)}) = q. \tag{4.49}$$

Si $q < 0$ et si λ est défini par les conditions

$$f'_{21}(\lambda) \leqslant q \leqslant f'_{22}(\lambda), \tag{4.50}$$

alors

$$r(q) = R(\alpha X_{\lambda 1}^{(2)} + (1-\alpha) X_{\lambda 2}^{(2)}, \tag{4.51}$$

où le coefficient α se calcule d'après la relation

$$\alpha L_2(X_{\lambda 1}^{(2)}) + (1-\alpha) L_2(X_{\lambda 2}^{(2)}) = q. \tag{4.52}$$

Démonstration. 1. Puisque le dénominateur de la fraction $R(X)$ est fixé et constitue un nombre positif, pour justifier (4.48) il suffit de s'assurer que le vecteur

$$X_{\lambda\alpha}^{(1)} = \alpha X_{\lambda 1}^{(1)} + (1-\alpha) X_{\lambda 2}^{(1)},$$

où λ et α sont déterminés respectivement d'après (4.47) et (4.49), est une solution du problème de maximisation de $L_1(X)$ sous les conditions

$$X \in M, \quad L_2(X) = q. \tag{4.53}$$

Les relations (4.46), (4.47) et (4.49) conduisent aux inégalités

$$0 \leqslant \alpha \leqslant 1.$$

Par conséquent, $X_{\lambda\alpha}^{(1)}$ est une solution du problème de maximisation de $L_\lambda(X)$ sous les contraintes (4.41), (4.42), du fait que cette condition est vérifiée par les vecteurs $X_{\lambda 1}^{(1)}$ et $X_{\lambda 2}^{(1)}$. Il est maintenant facile d'établir la propriété d'extrémum cherchée du vecteur $X_{\lambda\alpha}^{(1)}$ à l'aide des raisonnements utilisés lors de la démonstration du lemme 4.2 (se servir à cette fin du critère d'optimalité du programme $X_{\lambda\alpha}^{(1)}$).

2. Soit $q < 0$. L'égalité (4.51) est justifiée par des raisonnements analogues à ceux qui ont été donnés plus haut; il n'y a qu'à remarquer que lorsque la fraction $R(X)$ a un dénominateur fixé négatif, son numérateur doit être aussi petit que possible. Le lemme est ainsi démontré.

Soit λ une valeur quelconque du paramètre tel que $f'_{11}(\lambda) > 0$. Introduisons l'ensemble $E_\lambda^{(1)}$ des nombres $q > 0$ qui vérifient les contraintes (4.47). Si λ n'est pas une valeur critique de la fonction $f_1(\lambda)$, l'ensemble $E_\lambda^{(1)}$ se compose du seul point

$$q = f'_{11}(\lambda) = f'_{12}(\lambda).$$

Dans le cas contraire, $E_\lambda^{(1)}$ est soit un segment $[f'_{12}(\lambda), f'_{11}(\lambda)]$ lorsque $f'_{12}(\lambda) > 0$, soit un demi-intervalle $(0, f'_{11}(\lambda)]$, lorsque $f'_{12}(\lambda) \leqslant 0$.

De façon analogue, introduisons pour un point quelconque λ tel que $f'_{21}(\lambda) < 0$, l'ensemble $E_\lambda^{(2)}$ composé des nombres $q < 0$ qui vérifient les inégalités (4.50). L'ensemble $E_\lambda^{(2)}$ est lui aussi soit un point, soit un segment, soit encore un demi-intervalle.

Lemme 4.5. *Si l'ensemble* $E_\lambda^{(1)}$ *est défini, avec* $f_1(\lambda) > 0$, $f_1(\lambda) < 0$, $f_1(\lambda) = 0$, *la fonction* $r(q)$ *est respectivement monotone décroissante, monotone croissante, à valeur constante sur* $E_\lambda^{(1)}$.

Si l'ensemble $E_\lambda^{(2)}$ *est défini, avec* $f_2(\lambda) > 0$, $f_2(\lambda) < 0$, $f_2(\lambda) = 0$, *la fonction* $r(q)$ *est respectivement monotone décroissante, monotone croissante, à valeur constante sur* $E_\lambda^{(2)}$.

Démonstration. Supposons que l'ensemble $E_\lambda^{(1)}$ contienne plus d'un point et que de plus, q soit un point intérieur de $E_\lambda^{(1)}$. D'après le lemme 4.4,

$$r(q) = R(X_{\lambda\alpha}^{(1)}),$$

où

$$X_{\lambda\alpha}^{(1)} = \alpha X_{\lambda_1}^{(1)} + (1-\alpha) X_{\lambda_2}^{(1)}$$

est une solution du problème de maximisation de la fonction $L_\lambda(X)$ sous les conditions (4.41), (4.42); $L_2(X_{\lambda\alpha}^{(1)}) = q$. Par conséquent,

$$f_1(\lambda) = L_1(X_{\lambda\alpha}^{(1)}) + \lambda L_2(X_{\lambda\alpha}^{(1)}), \tag{4.54}$$

$$r(q) = R(X_{\lambda\alpha}^{(1)}) = \frac{f_1(\lambda)}{q} - \lambda.$$

En dérivant (4.54) par rapport à $q \in E_\lambda^{(1)}$, on a

$$\frac{dr(q)}{dq} = -\frac{f_1(\lambda)}{q^2}. \tag{4.55}$$

Si q est un point intérieur de $E_\lambda^{(2)}$, les mêmes raisonnements impliquent

$$\frac{dr(q)}{dq} = -\frac{f_2(\lambda)}{q^2}. \tag{4.56}$$

Les formules (4.55) et (4.56) restent valables également pour les points frontières q, il suffit d'entendre par $\frac{dr(q)}{dq}$ la dérivée à gauche si q est l'extrémité droite de l'ensemble correspondant, et la dérivée à droite si q est son extrémité gauche.

Les relations (4.55) et (4.56) prouvent la validité du lemme.

4.7. Passons à l'énoncé et à la démonstration du théorème qui est à la base de l'algorithme de résolution du problème (4.40)-(4.42).

Théorème 4.1. 1. *Si* $q_{\min} < 0$, $q_{\max} > 0$, *une condition nécessaire et suffisante pour que le problème* (4.40)-(4.42) *soit résoluble est que chacune des fonctions* $f_1(\lambda)$ *et* $f_2(\lambda)$ *ait un seul point critique* λ^* *(commun aux deux fonctions) et, que, de plus,*

$$f_1(\lambda^*) = f_2(\lambda^*) = 0.$$

Dans le cas où le problème (4.40)-(4.42) *a une solution réalisable, tout point* $X \in M$, $L_2(X) \neq 0$ *en est une solution:*

$$r = R(X) = -\lambda^*.$$

2. *Si* $q_{\min} \geqslant 0$, *une condition nécessaire et suffisante pour que le problème* (4.40)-(4.42) *soit résoluble est que* $q_{\max}$ *soit un nombre positif et que la fonction* $f_1(\lambda)$ *ait des racines sur l'axe* λ. *Dans le cas où le problème* (4.40)-(4.42) *est résoluble, sa solution est fournie par le vecteur* $X_{s1}^{(1)}$, *où s est la plus grande racine de* $f_1(\lambda)$; *de plus,*

$$r = R(X_{s1}^{(1)}) = -s.$$

3. *Si* $q_{\max} \leqslant 0$, *une condition nécessaire et suffisante pour que le problème* (4.40)-(4.42) *soit résoluble est que* $q_{\min}$ *soit un nombre négatif et que la fonction* $f_2(\lambda)$ *ait des racines sur l'axe* λ.

Dans le cas où le problème (4.40)-(4.42) *est résoluble, sa solution est le vecteur* $X_{s1}^{(2)}$, *où s est la plus grande racine de* $f_2(\lambda)$; *de plus,*

$$r = R(X_{s1}^{(2)}) = -s.$$

Démonstration. 1. Soient $q_{\min} < 0$, $q_{\max} > 0$.

Condition nécessaire. Admettons que le problème (4.40)-(4.42) ait une solution réalisable. Dans ce cas, un point quelconque $X \in M$ tel que $L_2(X) = 0$ vérifie l'égalité

$$L_1(X) = 0. \tag{4.57}$$

Notons qu'il résulte des conditions du théorème qu'il existe des points $X' \in M$ et $X'' \in M$ tels que

$$L_2(X') > 0, \quad L_2(X'') < 0.$$

Avec $L_1(X) > 0$, le vecteur $X_\varepsilon = (1-\varepsilon)X + \varepsilon X'$, où $X' \in M$, $L_2(X') > 0$, appartient à M quel que soit $0 \leqslant \varepsilon \leqslant 1$, et

$$R(X_\varepsilon) = \frac{(1-\varepsilon)L_1(X) + \varepsilon L_1(X')}{\varepsilon L_2(X')} \to \infty, \text{ si } \varepsilon \to 0.$$

La relation obtenue réduit à l'absurde notre hypothèse sur l'existence d'un programme. Par conséquent, $L_1(X) \leqslant 0$. On aboutit de façon analogue à l'autre contrainte : $L_1(X) \geqslant 0$. L'égalité (4.57) est ainsi démontrée.

Considérons deux points quelconques X' et X'' du polyèdre M tels que

$$L_2(X') > 0, \quad L_2(X'') < 0.$$

Soit

$$L_1(X') = k'L_2(X'), \quad L_1(X'') = k''L_2(X'').$$

Montrons que

$$k' = k'' = k. \tag{4.58}$$

En effet, posons que le point $X = \lambda X' + (1-\lambda)X'' \in M$ est défini par la condition

$$L_2(X) = 0. \tag{4.59}$$

Il est évident que $0 < \lambda < 1$. D'après (4.57),

$$L_1(X) = \lambda k' L_2(X') + (1 - \lambda) k'' L_2(X'') = 0.$$

Transformons cette dernière égalité en tenant compte de (4.59):

$$L_1(X) = \lambda L_2(X')(k' - k'') = 0.$$

Nous avons déjà dit que

$$\lambda > 0, \quad L_2(X') > 0.$$

Donc, $k' - k'' = 0$. L'égalité (4.58) est démontrée.

Il est facile de vérifier maintenant que tout point $X \in M$ justifie l'égalité

$$L_1(X) = kL_2(X), \tag{4.60}$$

où k ne dépend pas de X.

En effet, avec $L_2(X) > 0$ ($L_2(X) < 0$), il suffit de considérer les points $X' = X$ et X'' (X' et $X'' = X$) et d'utiliser (4.58). Mais si $L_2(X) = 0$, (4.60) se déduit alors de (4.57).

Si λ est un nombre arbitraire et X un point quelconque du polyèdre M, alors en vertu de (4.60),

$$L_\lambda(X) = L_1(X) + \lambda L_2(X) = L_2(X)[k + \lambda]. \tag{4.61}$$

La relation (4.61) amène les égalités

$$\left.\begin{aligned} f_1(\lambda) &= \begin{cases} q_{\max}(k + \lambda) & \text{si } \lambda \geqslant -k, \\ q_{\min}(k + \lambda) & \text{si } \lambda < -k; \end{cases} \\ f_2(\lambda) &= \begin{cases} q_{\min}(k + \lambda) & \text{si } \lambda \geqslant -k, \\ q_{\max}(k + \lambda) & \text{si } \lambda < -k. \end{cases} \end{aligned}\right\} \tag{4.62}$$

La condition nécessaire de la première partie du théorème est ainsi démontrée, et $\lambda^* = -k$.

Condition suffisante. Supposons que les conditions de la première partie du théorème soient remplies. Il est clair qu'avec $\lambda^* = -k$ les fonctions $f_1(\lambda)$ et $f_2(\lambda)$ doivent avoir la forme (4.62).

Pour tout $X \in M$, on a

$$0 = f_2(-k) \leqslant L_{-k}(X) \leqslant f_1(-k) = 0,$$

et

$$L_{-k}(X) = L_1(X) - kL_2(X) = 0.$$

Par conséquent, les conditions

$$X \in M, \quad L_2(X) \neq 0 \tag{4.63}$$

impliquent l'égalité

$$R(X) = k = -\lambda^*,$$

ce qui signifie que le problème (4.40)-(4.42) a un programme et que, de plus, tout point vérifiant les conditions (4.63) en est une solution.

2. Soit $q_{\min} \geqslant 0$.

Condition nécessaire. Supposons que le problème (4.40)-(4.42) ait une solution réalisable. Si $q_{\max} = 0$, $q_{\min} = 0$ et, par conséquent,

$$L_2(X) = 0 \quad \text{pour tout} \quad X \in M,$$

ce qui signifie que la fonction $R(X)$ n'est définie en aucun point du polyèdre M. Cette contradiction entraîne la positivité de $q_{\max}$.

Pour démontrer que l'équation $f_1(\lambda) = 0$ a une solution, nous procéderons également par la réduction à l'absurde. Supposons que cette équation n'ait aucune racine. Pour des valeurs assez grandes de l'argument $f_1(\lambda)$ est une fonction linéaire à coefficient angulaire $q_{\max} > 0$. Il résulte donc de l'hypothèse adoptée que $f_1(\lambda)$ est positive pour n'importe quelle valeur de λ. Constatons également que $q_{\min} = 0$, puisque avec $q_{\min} > 0$ la fonction $f_1(\lambda)$ couperait obligatoirement l'axe λ.

Soit s la valeur critique minimale de la fonction $f_1(\lambda)$. Il est évident que dans ce cas

$$L_2(X_{s2}^{(1)}) = q_{\min} = 0, \quad L_2(X_{s1}^{(1)}) > 0, \tag{4.64}$$

et de plus,

$$L_s(X_{s1}^{(1)}) = L_s(X_{s2}^{(1)}) = f_1(s) > 0. \tag{4.65}$$

Posons

$$X_\varepsilon = (1 - \varepsilon) X_{s2}^{(1)} + \varepsilon X_{s1}^{(1)}.$$

Quel que soit $0 \leqslant \varepsilon \leqslant 1$, le point $X_\varepsilon \in M$. Les relations (4.64) et (4.65) donnent

$$R(X_\varepsilon) = \frac{f_1(s)}{L_2(X_\varepsilon)} - s \to \infty \quad \text{quand} \quad \varepsilon \to 0,$$

ce qui contredit l'existence d'une solution réalisable du problème (4.40)-(4.42). Par conséquent, notre hypothèse est incorrecte, ce qui signifie que la fonction $f_1(\lambda)$ a des racines.

Condition suffisante. Supposons que la fonction $f_1(\lambda)$ ait des racines sur l'axe des λ et que $q_{\max} > 0$. Désignons par s la racine maximale de $f_1(\lambda)$.

Posons

$$q' = L_2(X_{s2}^{(1)}), \quad q'' = L_2(X_{s1}^{(1)}).$$

s étant la racine maximale de $f_1(\lambda)$, $q'' > 0$, et donc l'ensemble $E_s^{(1)}$ est défini.

En vertu du lemme 4.5,

$$r(q) = r(q'') \quad \text{pour} \quad q \in E_s^{(1)}, \tag{4.66}$$

puisque $f_1(s) = 0$.

Supposons que $q'' < q_{\max}$. Le segment $[q'', q_{\max}]$ peut être partitionné en segments $E_{\lambda_i}^{(1)}$, $i = 1, 2, \ldots, p$, où $\lambda_1, \lambda_2, \ldots, \lambda_p$ est une collection complète de valeurs critiques de la fonction $f_1(\lambda)$ à droite du point s. Par condition, $f_1(\lambda_i) > 0$, puisque $\lambda_i > s$. Par suite, d'après le lemme 4.5, la fonction $r(q)$ est monotone décroissante sur tout segment $E_{\lambda_i}^{(1)}$. Autrement dit,

$$r(q) < r(q'') \quad \text{si} \quad q'' < q \leqslant q_{\max}. \tag{4.67}$$

Supposons que $q' > q_{\min}$. L'ensemble des points $q > 0$ entre $q_{\min}$ et q' peut être partitionné en ensembles $E_{\lambda_i}^{(1)}$, $i = -1, -2, \ldots$ $\ldots, -t$, où $\lambda_{-1}, \lambda_{-2}, \ldots, \lambda_{-t}$ est une collection complète des valeurs critiques de la fonction $f_1(\lambda)$ à gauche du point s. Etant donné que $q' > q_{\min} \geqslant 0$,

$$f_1(\lambda) < f_1(s) = 0 \quad \text{si} \quad \lambda < s.$$

Donc,

$$f_1(\lambda_{-i}) < 0, \quad i = -1, -2, \ldots, -t,$$

et, d'après le lemme 4.5, la fonction $r(q)$ est monotone croissante sur tout ensemble E_{λ_i} $(i = -1, -2, \ldots, -t)$.

C'est pourquoi

$$r(q) < r(q') \quad \text{si} \quad q > 0, \quad q_{\min} \leqslant q < q'. \tag{4.68}$$

Les relations (4.66)-(4.68) mènent à l'inégalité

$$R(X_{s1}^{(1)}) = r(q'') \geqslant r(q), \quad \text{si} \quad q > 0, \quad q_{\min} \leqslant q \leqslant q_{\max},$$

qui peut être mise sous la forme

$$r = \sup_{X \in M} R(X) = \sup_{q_{\min} \leqslant q \leqslant q_{\max},\, q > 0} r(q) = R(X_{s1}^{(1)}).$$

Donc, $X_{s1}^{(1)}$ est la solution recherchée du problème.

Etant donné que

$$L_s(X_{s1}^{(1)}) = f_1(s) = 0,$$

il vient

$$R(X_{s1}^{(1)}) = -s.$$

La deuxième partie du théorème est également démontrée.

3. La démonstration de la dernière partie du théorème ne diffère des raisonnements de la deuxième partie que par la substitution de $f_2(\lambda)$ à $f_1(\lambda)$.

4.8. Voici la description de l'algorithme de résolution du problème (4.40)-(4.42), fondé sur la méthode de programmation paramétrique exposée au § 1.

Désignons par $A_{\lambda 1}$ ($A_{\lambda 2}$) le problème qui consiste à maximiser (minimiser) la fonction $L_\lambda(X)$ sous les conditions (4.41), (4.42).

1. Cherchons la solution X_1 du problème de maximisation de la fonction $L_2(X)$ sous les conditions (4.41), (4.42). Deux cas sont possibles :

$$\text{a)}\quad q_{\max} = L_2(X_1) > 0; \quad \text{b)}\quad q_{\max} = L_2(X_1) \leqslant 0.$$

2.a. Soit $L_2(X_1) > 0$. Appliquons la méthode du § 1 pour étudier le problème paramétrique $A_{\lambda 1}$.

Prenons comme programme initial le programme X_1, qui est une solution du problème $A_{\lambda 1}$ avec des valeurs de λ suffisamment grandes. En cours d'analyse, les valeurs du paramètre λ sont donc astreintes à décroître. On sait que les calculs, dans ce cas, consistent à effectuer plusieurs itérations de la méthode primale du simplexe telles que le choix du vecteur A_k à ajouter à la base soit défini par les conditions

$$-\frac{\Delta_k^{(1)}}{\Delta_k^{(2)}} = \max_{\Delta_j^{(2)}>0}\left(-\frac{\Delta_j^{(1)}}{\Delta_j^{(2)}}\right), \quad \Delta_k^{(2)} > 0, \tag{4.69}$$

où $\Delta_j^{(1)}$ et $\Delta_j^{(2)}$ sont les critères relatifs au vecteur A_j associés aux formes linéaires $L_1(X) - c$ et $L_2(X) - d$ respectivement.

Il se peut que X_1 soit une solution du problème $A_{\lambda 1}$ pour toutes les valeurs de λ. Alors, d'après la deuxième partie du théorème 4.1, le vecteur X_1 est une solution du problème (4.40)-(4.42). S'il n'en est pas ainsi, l'axe λ est découpé en ensembles $\delta_{l+1} = (-\infty, \lambda_l]$, $\delta_l = [\lambda_l, \lambda_{l-1}]$, . . ., $\delta_2 = [\lambda_2, \lambda_1]$, $\delta_1 = [\lambda_1, \infty)$; de plus, on construit pour tout $1 \leqslant i \leqslant l+1$ un programme de base X_i qui est une solution du problème $A_{\lambda 1}$ avec $\lambda \in \delta_i$. La marche ultérieure de la procédure est déterminée par le signe de $q_{\min} = L_2(X_{l+1})$.

3.a.a. Soit $L_2(X_{l+1}) < 0$. Nous sommes alors en présence des conditions de la première partie du théorème 4.1. Pour rendre le problème possible, il faut justifier les restrictions

$$\lambda_l = \lambda_1, \quad L_{\lambda_1}(X_1) = L_{\lambda_1}(X_2) = 0. \tag{4.70}$$

Si les égalités (4.70) ne sont pas respectées, le problème (4.40)-(4.42) n'a pas de solutions réalisables. Mais si les relations (4.70) ont lieu, on vérifie alors les conditions

$$\left.\begin{array}{l} X_1 \text{ est une solution du problème } A_{\lambda_2} \text{ pour } \lambda \leqslant \lambda_1, \\ X_2 \text{ est une solution du problème } A_{\lambda_2} \text{ pour } \lambda \geqslant \lambda_1. \end{array}\right\} \tag{4.71}$$

Lorsque ces conditions sont vérifiées, cela signifie que le problème (4.40)-(4.42) est possible et que le vecteur X_1 en donne la solution.

Dans le cas où les restrictions (4.71) ne sont pas observées, on en tire la conclusion que le problème n'a pas de solution réalisable.

3.a.b. Soit $L_2(X_{l+1}) \geqslant 0$. Dans ce cas, on achève la discussion du problème à l'aide de la deuxième partie du théorème 4.1. La fonction $f_1(\lambda)$ est alors non décroissante. Donc si

$$f_1(\lambda_l) = L_{\lambda_l}(X_l) > 0, \quad L_2(X_{l+1}) = 0, \tag{4.72}$$

$f_1(\lambda)$ n'a pas de racines sur l'axe des λ, ce qui signifie que le problème (4.40)-(4.42) est irrésoluble.

Mais si les conditions (4.72) ne sont pas vérifiées, on choisit l'indice α maximal tel que

$$f_1(\lambda_\alpha) = L_{\lambda_\alpha}(X_\alpha) \geqslant 0.$$

D'après le théorème 4.1, le vecteur

$$X^* = \begin{cases} X_{\alpha+1} & \text{si } f_1(\lambda_\alpha) > 0, \\ X_\alpha & \text{si } f_1(\lambda_\alpha) = 0 \end{cases}$$

est la solution cherchée du problème (4.40)-(4.42).

2.b. Supposons maintenant que $q_{\max} \leqslant 0$. Ce cas correspond à la dernière partie du théorème 4.1. Il faut explorer le problème paramétrique $A_{\lambda 2}$. Etant donné que X_1 est une solution pour des valeurs suffisamment petites du paramètre λ, le choix de X_1 comme programme initial assure une croissance monotone du paramètre λ au cours de la discussion du problème $A_{\lambda 2}$. Les calculs consistent également à effectuer plusieurs itérations de la méthode primale du simplexe, le vecteur A_k à inclure dans la base se détermine de nouveau d'après la relation (4.69).

Il peut s'avérer que X_1 soit un programme optimal du problème $A_{\lambda 2}$ avec tout λ. Si dans ces conditions $q_{\max} = q_{\min} = 0$, le problème (4.40)-(4.42) n'a pas de solution du fait que la fonction $R(X)$ n'est définie en aucun point du polyèdre M. Si, par contre, $q_{\max} = q_{\min} < 0$, d'après la troisième partie du théorème 4.1, X_1 est une solution du problème (4.40)-(4.42).

Dans le cas général, l'axe λ est découpé en ensembles

$$\delta_1' = (-\infty, \lambda_1], \quad \delta_2' = [\lambda_1, \lambda_2], \ldots, \delta_p' = [\lambda_{p-1}, \lambda_p], \quad \delta_{p+1}' = [\lambda_p, \infty)$$

et l'on construit pour tout i $(1 \leqslant i \leqslant p+1)$ un programme optimal X_i du problème $A_{\lambda 2}$ avec $\lambda \in \delta_i'$.

3.b. La fonction $f_2(\lambda)$ est non croissante, puisque $q_{\max} \leqslant 0$. C'est pourquoi, les conditions

$$f_2(\lambda_1) = L_{\lambda_1}(X_1) < 0, \quad q_{\max} = L_2(X_1) = 0 \tag{4.73}$$

étant remplies, $f_2(\lambda)$ n'a pas de racines sur l'axe λ, ce qui indique que le problème (4.40)-(4.42) n'a donc pas de solution réalisable.

Supposons que les conditions (4.73) ne soient pas respectées. Choisissons l'indice α maximal tel que $f_2(\lambda_\alpha) = L_{\lambda_\alpha}(X_\alpha) = L_{\lambda_\alpha}(X_{\alpha+1}) \geqslant 0$. D'après le théorème 4.1, le vecteur $X_{\alpha+1}$ est la solution cherchée du problème (4.40)-(4.42).

Ainsi, la méthode de programmation paramétrique exposée au § 1 permet d'effectuer une exploration complète du problème de programmation homographique.

Insistons sur le fait que l'algorithme ainsi décrit pour la résolution du problème (4.40)-(4.42) se compose d'itérations de la méthode primale du simplexe appliquée au problème linéaire de conditions (4.41), (4.42). Pour cette raison, si par exemple (4.41), (4.42) sont les contraintes d'un problème de transport, l'algorithme ne diffère que de peu de la méthode des potentiels. Dans certains cas, les conditions (4.41), (4.42) sont complétées par des contraintes sur le dénominateur $L_2(X)$ de la fraction $R(X)$ du type $\beta_1 \leqslant L_2(X) \leqslant \beta_2$. Il est facile de remarquer (cf. exercice 22) que la présence de ces conditions supplémentaires ne complique nullement l'algorithme.

C'est pourquoi lorsque le système (4.41), (4.42) contient de telles conditions, on aura intérêt à les mettre en évidence. Pour finir, indiquons une particularité qualitative des problèmes du type (4.40)-(4.42). La fonction (4.40) ne jouit pas de la propriété de convexité, si bien que le problème (4.40)-(4.42) peut avoir un maximum local distinct du maximum global*). Toutefois, la structure simple de la fonction $R(X)$ aboutit à ce que cette éventualité qui, d'une façon générale, est cause de difficultés de calcul considérables, est dans le cas présent tout à fait « sans danger ». Le fait est qu'en présence d'un maximum local du problème (4.40)-(4.42) qui ne coïncide pas avec son maximum global, la face supérieure de $R(X)$ sur le polyèdre M est infinie (cf. exercice 23). C'est, au fond, cette circonstance qui a fourni la possibilité de construire un algorithme assez simple de résolution du problème (4.40)-(4.42).

§ 5. Sensibilité des solutions des problèmes de programmation linéaire à la variation des données

5.1. La question de la vitesse de variation de la valeur optimale de la forme linéaire d'un problème en fonction de la variation des composantes des vecteurs des conditions, des composantes du vecteur des contraintes et des coefficients de la forme linéaire présente un grand intérêt pour les applications. L'étude des dérivées de la valeur

*) $R(X')$ est un maximum local du problème (4.40)-(4.42) si $R(X) \leqslant R(X')$ avec $X \in M$, $L_2(X) \neq 0$, $|X - X'| \leqslant \delta$, δ est un certain nombre positif. Par maximum global du problème, on entend $\sup\limits_{X \in M} R(X)$.

optimale de la forme linéaire par rapport aux a_{ij}, b_i, c_j et à leurs combinaisons linéaires, de même que la discussion des problèmes de programmation linéaire, permet d'apprécier la stabilité de la solution d'un problème dans les conditions des variations possibles des éléments de sa matrice complète des conditions.

Considérons le problème possible de programmation linéaire A qui consiste à maximiser la forme linéaire

$$\sum_{j=1}^{n} c_j x_j \tag{5.1}$$

sous les conditions

$$\sum_{j=1}^{n} a_{ij} x_j = b_i, \quad i = 1, 2, \ldots, m, \tag{5.2}$$

$$x_j \geqslant 0, \quad j = 1, 2, \ldots, n. \tag{5.3}$$

Désignons par

$$f(a_{ij}, c_j, b_i)$$

la valeur maximale de la forme linéaire (5.1) sous les conditions (5.2), (5.3), associée, naturellement, à tous les éléments a_{ij}, c_j, b_i ($i = 1, 2, \ldots, m$; $j = 1, 2, \ldots, n$) de la matrice complète des conditions du problème A.

S'imposant une certaine direction $\Delta = \{\delta_{ij}, \delta_{0j}, \delta_{i0}\}$ de variation des paramètres a_{ij}, c_j, b_i du problème A, introduisons la dérivée $f'_\Delta(a_{ij}, c_j, b_i)$ de la fonction $f(a_{ij}, c_j, b_i)$ le long de la direction Δ. Conformément à la définition connue de la dérivée par rapport à la direction

$$f'_\Delta(a_{ij}, c_j, b_i) = \varphi'_\Delta(0) = \lim_{t \to +0} \frac{\varphi(t) - \varphi(0)}{t}, \tag{5.4}$$

où

$$\varphi(t) = f(a_{ij} + t\delta_{ij}, c_j + t\delta_{0j}, b_i + t\delta_{i0}).$$

Il est logique de supposer à cet effet que le problème $A(t)$ dont la matrice complète des conditions se compose des éléments $a_{ij} + t\delta_{ij}$, $c_j + t\delta_{0j}$ et $b_i + t\delta_{i0}$ ($i = 1, 2, \ldots, m$; $j = 1, 2, \ldots, n$) ait une solution pour $t \in [0, t_0]$, $t_0 > 0$, et que la relation (5.4) ait une limite.

Les dérivées f'_Δ caractérisent l'influence des fluctuations des paramètres du problème sur la valeur optimale de sa forme linéaire et sont souvent utilisées dans les applications.

Le présent paragraphe est consacré à l'établissement des relations qui associent les valeurs des dérivées f'_Δ aux solutions et aux critères du problème $A = A(0)$. Les formules données ci-dessous permettent de calculer les dérivées f'_Δ sans explorer au préalable le problème $A(t)$ pour $t > 0$ et ont une grande valeur pratique.

5.2. Composons la fonction de Lagrange $F_t(X, Y)$ pour le problème $A(t)$ aux paramètres

$$a_{ij}(t) = a_{ij} + t\delta_{ij}, \quad a_{0j}(t) = c_j + t\delta_{0j}, \quad a_{i0}(t) = b_i + t\delta_{i0}$$
$$(i = 1, 2, \ldots, m; \quad j = 1, 2, \ldots, n),$$

$$F_t(X, Y) = \sum_{j=1}^{n} a_{0j}(t)\, x_j + \sum_{i=1}^{m} a_{i0}(t)\, y_i - \sum_{i=1}^{m} \sum_{j=1}^{n} a_{ij}(t)\, y_i x_j, \quad (5.5)$$

Pour le problème initial $A = A(0)$ défini par les relations (5.1)-(5.3) la fonction de Lagrange s'écrit $F(X, Y) = F_0(X, Y)$.

Nous aurons besoin par la suite de la notion de point selle, qui généralise la définition donnée dans le chapitre 2 en vue d'être appliquée aux jeux (cf. également le paragraphe 4.4 du chapitre 6 de [87]).

Appelons $(X_0, Y_0) \in T_X \times T_Y$ point selle de la fonction $R(X, Y)$, sous la condition $(X, Y) \in T_X \times T_Y$, si les relations

$$R(X, Y_0) \leqslant R(X_0, Y_0) \leqslant R(X_0, Y) \quad (5.6)$$

ont lieu pour tout point $(X, Y) \in T_X \times T_Y$. D'après le théorème 4.6 du chapitre 6 de [87], les vecteurs X_t et Y_t sont respectivement solutions du problème $A(t)$ et du problème $\widetilde{A}(t)$ dual de $A(t)$ si et seulement si (X_t, Y_t) est un point selle de la fonction $F_t(X, Y)$ sous la condition: $X_t \geqslant 0$, Y_t étant quelconque. On voit aisément que, de plus,

$$F_t(X_t, Y_t) = \sum_{j=1}^{n} a_{0j}(t)\, x_j^{(t)} = \sum_{i=1}^{m} a_{i0}(t)\, y_i^{(t)} = \varphi(t). \quad (5.7)$$

Introduisons dans la discussion la fonction

$$\Gamma(X, Y) = \sum_{j=1}^{n} \delta_{0j} x_j + \sum_{i=1}^{m} \delta_{i0} y_i - \sum_{i=1}^{m} \sum_{j=1}^{n} \delta_{ij} y_i x_j, \quad (5.8)$$

engendrée par la direction donnée $\Delta = \{\delta_{ij}, \delta_{0j}, \delta_{i0}\}$.

Avant d'énoncer la proposition principale de ce paragraphe, nous allons démontrer une proposition concernant les points selle de la fonction $\Gamma(X, Y)$.

L e m m e 5.1. *Si G_X et G_Y sont des polyèdres convexes, la fonction $\Gamma(X, Y)$ a un point selle (X^*, Y^*) sous la condition que $(X, Y) \in G_X \times G_Y$; de plus,*

$$\max_{X \in G_X} \min_{Y \in G_Y} \Gamma(X, Y) = \min_{Y \in G_Y} \max_{X \in G_X} \Gamma(X, Y) = \Gamma(X^*, Y^*). \quad (5.9)$$

D é m o n s t r a t i o n. Soit X_k, $k = 1, 2, \ldots, N_1$, la collection complète des sommets du polyèdre G_X, et Y_k, $k = 1, 2, \ldots, N_2$, l'ensemble complet des sommets du polyèdre G_Y. D'après

le théorème connu sur la représentation des polyèdres convexes (théorème 2.1 du chapitre 5 de [87], les polyèdres G_X et G_Y coïncident avec les ensembles de points

$$X=\sum_{k=1}^{N_1}\alpha_k X_k,\quad \sum_{k=1}^{N_1}\alpha_k=1,\quad \alpha_k\geqslant 0, \tag{5.10}$$

et

$$Y=\sum_{k=1}^{N_2}\beta_k Y_k,\quad \sum_{k=1}^{N_2}\beta_k=1,\quad \beta_k\geqslant 0, \tag{5.11}$$

respectivement.

En transformant l'expression (5.8) pour $\Gamma(X, Y)$ à l'aide des relations (5.10) et (5.11), on a

$$\Gamma(X, Y)=\sum_{k=1}^{N_1}\alpha_k d_{0k}+\sum_{s=1}^{N_2}\beta_s d_{s0}-\sum_{k=1}^{N_1}\sum_{s=1}^{N_2}\alpha_k\beta_s d_{sk}, \tag{5.12}$$

$$\sum_{k=1}^{N_1}\alpha_k=1,\quad \sum_{s=1}^{N_2}\beta_s=1,\quad \alpha_k\geqslant 0,\ \beta_s\geqslant 0, \tag{5.13}$$

où

$$d_{0k}=\sum_{j=1}^{n}\delta_{0j}x_j^{(k)};\quad d_{s0}=\sum_{i=1}^{m}\delta_{i0}y_i^{(s)};\quad d_{sk}=\sum_{i=1}^{m}\sum_{j=1}^{n}\delta_{ij}y_i^{(s)}x_j^{(k)}.$$

Si l'on utilise maintenant les conditions (5.13), la relation (5.12) peut se mettre sous la forme bilinéaire homogène

$$\Gamma(X, Y)=H(\alpha, \beta)=\sum_{k=1}^{N_1}\sum_{s=1}^{N_2}h_{sk}\alpha_k\beta_s,$$

où

$$h_{sk}=d_{0k}+d_{s0}-d_{sk}\ (k=1, 2, \ldots, N_1;\ s=1, 2, \ldots, N_2).$$

Ainsi, la recherche d'un point selle de la fonction $\Gamma(X, Y)$ sous la condition $(X, Y)\in G_X\times G_Y$ se ramène à trouver un point selle de la fonction $H(\alpha, \beta)$ sous la condition (5.13), ce qui est équivalent à la résolution d'un jeu à matrice $\|h_{sk}\|$. C'est pourquoi la démonstration du lemme se fonde sur le théorème fondamental des jeux matriciels (théorème 2.1, chapitre 2).

Désignons respectivement par G_X et G_Y les ensembles des solutions d'un couple de problèmes duals respectifs $A=A(0)$ et $\widetilde{A}=\widetilde{A}(0)$. Voici la formulation du théorème principal de ce paragraphe.

Théorème 5.1. *Si G_X et G_Y sont des ensembles bornés non vides, il existe une dérivée f'_Δ par rapport à une direction quelconque*

Δ, *telle qu'elle donne lieu à la relation*

$$f_\Delta = \max_{X \in G_X} \min_{Y \in G_Y} \Gamma(X, Y) = \min_{Y \in G_Y} \max_{X \in G_X} \Gamma(X, Y) = \Gamma(X^*, Y^*), \tag{5.14}$$

où (X^*, Y^*) *est un point selle de la fonction* $\Gamma(X, Y)$ *sous la condition* $X \in G_X$, $Y \in G_Y$.

La démonstration du théorème 5.1 s'appuie, en plus du lemme 5.1, sur trois autres propositions auxiliaires, dont l'énoncé et la justification sont donnés dans le paragraphe suivant.

5.3. Considérons une famille d'ensembles convexes non vides $M(t)$, $0 \leqslant t \leqslant t_1$, $t_1 > 0$, dont le système des conditions dépend linéairement du paramètre t:

$$\begin{aligned} (A^{(i)} + tA_1^{(i)}, Z) &\leqslant d_i + td_i^{(1)}, \quad i = 1, 2, \ldots l; \\ (A^{(i)} + tA_1^{(i)}, Z) &= d_i + td_i^{(1)}, \quad i = l+1, \ldots, l+t. \end{aligned} \tag{5.15}$$

$A^{(i)}$, $A_1^{(i)}$ sont ici des vecteurs-lignes, dont la dimension coïncide avec celle du vecteur-colonne Z;

d_i, $d_i^{(1)}$ sont des scalaires.

Lemme 5.2. *Si* $M(0)$ *est un ensemble borné (polyèdre convexe), il existe un* $t_0 > 0$ *tel qu'avec* $0 \leqslant t \leqslant t_0$ *les ensembles convexes* $M(t)$ *soient bornés dans leur généralité, c'est-à-dire*

$$\sup_{Z \in M(t),\ 0 \leqslant t \leqslant t_0} \|Z\| < \infty. \tag{5.16}$$

Démonstration. Supposons que (5.16) soit incorrect, c'est-à-dire qu'il existe des suites $t_k \to 0$ et $Z_k \in M(t_k)$ telles que

$$\| Z_k \| \to \infty. \tag{5.17}$$

Posons

$$\overline{Z}_k = \frac{Z_k}{|Z_k|}.$$

Prélevons dans la suite bornée $\{\overline{Z}_k\}$ la sous-suite convergente

$$Z_{k_\alpha} \to Z_0. \tag{5.18}$$

En divisant par $|Z_k|$ les deux membres des relations (5.15) pour $t = t_k$ et $Z = Z_k$, il vient

$$\left.\begin{aligned} (A^{(i)} + t_k A_1^{(i)}, \overline{Z}_k) &\leqslant \frac{d_i + t_k d_i^{(1)}}{|Z_k|}, \quad i = 1, 2, \ldots, l; \\ (A^{(i)} + t_k A_1^{(i)}, \overline{Z}_k) &= \frac{d_i + t_k d_i^{(1)}}{|Z_k|}, \quad i = l+1, \ldots, l+s. \end{aligned}\right\} \tag{5.19}$$

Faisons tendre vers l'infini l'indice k de (5.19) par rapport à la suite k_α. Compte tenu de (5.17) et (5.18), on obtient

$$\left.\begin{aligned} (A^{(i)}, Z_0) &\leqslant 0, \quad i = 1, 2, \ldots, l; \\ (A^{(i)}, Z_0) &= 0, \quad i = l+1, \ldots, l+s. \end{aligned}\right\} \tag{5.20}$$

Etant donné que $|Z_0| = |\overline{Z}_{k_\alpha}| = 1$, le vecteur $Z_0 \neq 0$.

Les relations (5.20) montrent que quels que soient

$$\lambda \geqslant 0 \quad \text{et} \quad Z \in M\,(0),$$

le point $Z + \lambda Z_0 \in M\,(0)$, c'est-à-dire l'ensemble $M\,(0)$ contient une demi-droite et ne peut, par conséquent, être borné. La contradiction ainsi obtenue est une preuve de la validité du lemme.

Lemme 5.3. *Si G_X et G_Y sont des ensembles bornés non vides, alors, quel que soit $\Delta = \{\delta_{ij}, \delta_{0j}, \delta_{i0}\}$, le problème $A\,(t)$ est résoluble pour tout t d'un certain voisinage à droite de zéro.*

Démonstration. Assurons-nous qu'il existe un $t_1 > 0$ tel que les conditions

$$\sum_{j=1}^{n} A_j(t)\,x_j = B(t), \quad x_j \geqslant 0, \quad j = 1, 2, \ldots, n, \tag{5.21}$$

du problème $A\,(t)$ soient compatibles quel que soit $t \in [0, t_1]$.

En effet, la supposition du contraire est équivalente à l'existence d'une suite $t_k \to 0$ jouissant de la propriété suivante: si $t = t_k$ $(k = 1, 2, \ldots)$, le cône polyédrique $K\,(t)$, constitué des points

$$Z = \sum_{j=1}^{n} A_j(t)\,x_j, \quad x_j \geqslant 0, \quad j = 1, 2, \ldots, n,$$

ne contient pas le point $B\,(t)$.

Appliquons au cône polyédrique $K\,(t_k)$ et au point $B\,(t_k) \notin K\,(t_k)$ le corollaire 3.2 du théorème de l'hyperplan séparateur (chapitre 4 de [87]). D'après ce corollaire, il se trouve un vecteur Λ_k $(|\Lambda_k| = 1)$ satisfaisant aux conditions

$$(\Lambda_k, Z) \geqslant 0 \quad \text{pour} \quad Z \in K\,(t_k), \tag{5.22}$$

$$(\Lambda_k, B\,(t_k)) < 0. \tag{5.23}$$

Les relations (5.22) sont équivalentes aux conditions

$$(\Lambda_k, A_j(t_k)) \geqslant 0, \quad j = 1, 2, \ldots, n. \tag{5.24}$$

Prélevons dans la suite bornée $\{\Lambda_k\}$ la sous-suite $\{\Lambda_{k_\alpha}\}$ convergeant vers un certain vecteur Λ $(|\Lambda| = 1)$.

En passant dans (5.23) et (5.24) à la limite avec $k = k_\alpha \to \infty$, il vient

$$(\Lambda, A_j) \geqslant 0, \quad j = 1, 2, \ldots, n; \tag{5.25}$$

$$(\Lambda, B) \leqslant 0. \tag{5.26}$$

Par condition, $B = B\,(0) \in K\,(0)$. Les relations (5.25) conduisent donc à l'inégalité

$$(\Lambda, B) \geqslant 0,$$

qui justifie avec (5.26) l'égalité

$$(\Lambda, B) = 0. \tag{5.27}$$

Soit Y une solution arbitraire du problème $\widetilde{A}$. Les relations (5.25) et (5.27) traduisent le fait que pour tout $\mu \geqslant 0$ le vecteur $Y + \mu\Lambda \in G_Y$. Le résultat obtenu est en contradiction avec l'hypothèse de l'ensemble borné G_Y. Cette dernière est donc fausse, ce qui signifie que pour $t \in [0, t_1]$, où t_1 est un certain nombre positif, le système (5.21) est compatible. On procède d'une façon analogue pour montrer la compatibilité des conditions du problème $\widetilde{A}(t)$ avec $t \in [0, t_2]$, $t_2 > 0$.

Ainsi, avec $t \in [0, t_3]$, $t_3 = \min(t_1, t_2) > 0$, les problèmes $A(t)$ et $\widetilde{A}(t)$ ont des programmes. D'après le corollaire 2.1 du premier théorème de dualité (ch. 6 de [87]), ceci indique que le problème $A(t)$ est résoluble pour $t \in [0, t_3]$, ce qui démontre le lemme 5.3.

Désignons par $\rho(Z_0, M)$ la distance entre le point Z_0 et l'ensemble fermé M, égale par définition à la face inférieure des distances entre le point Z_0 et les points $Z \in M$.

L e m m e 5.4. *Si les ensembles non vides G_X et G_Y des programmes optimaux des problèmes A et $\widetilde{A}$ respectivement sont bornés,*

$$\rho(G_X, X(t)) \to 0, \quad \rho(G_Y, Y(t)) \to 0 \tag{5.28}$$

avec $t \to 0$,
où $X(t)$ et $Y(t)$ sont respectivement les solutions des problèmes $A(t)$ et $\widetilde{A}(t)$.

D é m o n s t r a t i o n. On sait de la théorie de la dualité que l'ensemble $G(t) = G_X(t) \times G_Y(t)$ des solutions des problèmes $A(t)$ et $\widetilde{A}(t)$ peut être donné par le système de conditions linéaires

$$\left.\begin{aligned} &\sum_{j=1}^{n} a_{ij}(t)\, x_j = b_i(t), \quad i = 1, 2, \ldots, m; \\ &\qquad x_j \geqslant 0, \quad j = 1, 2, \ldots, n; \\ &\sum_{i=1}^{m} a_{ij}(t)\, y_i \geqslant c_j(t), \quad j = 1, 2, \ldots, n; \\ &\qquad \sum_{i=1}^{m} b_i(t)\, y_i = \sum_{j=1}^{n} c_j(t)\, x_j. \end{aligned}\right\} \tag{5.29}$$

D'après le lemme 5.3, l'ensemble $G(t)$ est non vide (c'est-à-dire que le système (5.29) est compatible) avec $t \in [0, t_3]$, $t_3 > 0$.

Les ensembles $G_X(0) = G_X$ et $G_Y(0) = G_Y$ étant bornés, il en résulte que l'ensemble $G(0) = G$ l'est également. Appliquons le

lemme 5.2 à la famille des polyèdres convexes $G(t)$, $0 \leqslant t \leqslant t_3$. La dépendance linéaire entre les coefficients des conditions (5.29) et le paramètre t entraîne que pour un certain $t_0 > 0$ les ensembles $G(t)$, $0 \leqslant t \leqslant t_0$ sont bornés dans leur généralité. Supposons maintenant que l'une au moins des relations (5.28) soit fausse. Admettons par exemple qu'il existe une suite $X_k \in G_X(t_k)$ telle que

$$\rho(G_X, X_k) > \varepsilon > 0, \quad k = 1, 2, \ldots, \tag{5.30}$$
$$\lim_{k\to\infty} t_k = 0, \quad t_k \leqslant t_0.$$

Posons

$$Z_k = (X_k, Y_k), \quad \text{où} \quad Y_k \in G_Y(t_k).$$

La suite des points $Z_k \in G(t_k)$, $0 < t_k \leqslant t_0$, est bornée. Prélevons-en une sous-suite convergente

$$Z_{k_\alpha} \to Z_0 = (X_0, Y_0).$$

En faisant tendre vers zéro t de la relation (5.29) par rapport à la suite t_{k_α}, $\alpha = 1, 2, \ldots$, après avoir préalablement substitué dans cette relation les vecteurs $X_{k_\alpha} \in G_X(t_{k_\alpha})$ et $Y_{k_\alpha} \in G_Y(t_{k_\alpha})$ à X et Y, on obtient

$$Z_0 = (X_0, Y_0) \in G_X \times G_Y.$$

Ainsi la suite $\{X_{k_\alpha}\}$ converge vers la solution X_0 du problème A, c'est-à-dire

$$\lim_{\alpha\to\infty} \rho(G_X, X_{k_\alpha}) = 0,$$

ce qui contredit (5.30). Le lemme 5.4 est ainsi démontré.

5.4. Passons à la justification du théorème 5.1.

Démonstration du théorème 5.1. La fonction $\varphi(t)$ est définie pour tout $t \in [0, t_0]$, $t_0 > 0$ (lemme 5.3); de plus, d'après la formule (5.7),

$$\varphi(t) = F_t(X_t, Y_t),$$

où X_t et Y_t sont des points quelconques des ensembles $G_X(t)$ et $G_Y(t)$ respectivement.

Par conséquent,

$$\frac{\varphi(t) - \varphi(0)}{t} = \frac{F_t(X_t, Y_t) - F_0(X_0, Y_0)}{t}, \quad 0 \leqslant t \leqslant t_0.$$

Conformément à la notation (5.8),

$$F_t(X, Y) = F_0(X, Y) + t\Gamma(X, Y).$$

Tenant compte du fait que (X_t, Y_t) est un point selle de la fonction $F_t(X, Y)$ et (X_0, Y_0) un point selle de la fonction $F_0(X, Y)$,

on a

$$F_t(X_t, Y_t) \geqslant F_t(X_0, Y_t) =$$
$$= F_0(X_0, Y_t) + t\Gamma(X_0, Y_t) \geqslant F_0(X_0, Y_0) + t\Gamma(X_0, Y_t), \quad (5.31)$$
$$F_t(X_t, Y_t) \leqslant F_t(X_t, Y_0) =$$
$$= F_0(X_t, Y_0) + t\Gamma(X_t, Y_0) \leqslant F_0(X_0, Y_0) + t\Gamma(X_t, Y_0). \quad (5.32)$$

Les inégalités (5.31) et (5.32) amènent la relation

$$\Gamma(X_0, Y_t) \leqslant \frac{\varphi(t) - \varphi(0)}{t} \leqslant \Gamma(X_t, Y_0). \quad (5.33)$$

D'après le lemme 5.4,

$$\lim_{t\to 0} \rho(X_t, G_X) = \lim_{t\to 0} \rho(Y_t, G_Y) = 0.$$

Par conséquent,

$$\left.\begin{aligned} &X_t = \overline{X}_t + \Delta X_t, \quad Y_t = \overline{Y}_t + \Delta Y_t, \\ &\text{où} \\ &\lim_{t\to 0} |\Delta X_t| = \lim_{t\to 0} |\Delta Y_t| = 0; \ \overline{X}_t \in G_X, \ \overline{Y}_t \in G_Y. \end{aligned}\right\} \quad (5.34)$$

La représentation (5.34) et le fait que les ensembles G_X et G_Y sont bornés permettent d'établir les critères suivants pour les nombres $\Gamma(X_0, Y_t)$ et $\Gamma(X_t, Y_0)$:

$$\left.\begin{aligned} &\Gamma(X_0, Y_t) \geqslant \Gamma(X_0, \overline{Y}_t) + \varepsilon_t' \geqslant \min_{Y \in G_Y} \Gamma(X_0, Y) + \varepsilon_t', \\ &\Gamma(X_t, Y_0) \leqslant \Gamma(\overline{X}_t, Y_0) + \varepsilon_t'' \leqslant \max_{X \in G_X} \Gamma(X, Y_0) + \varepsilon_t'', \end{aligned}\right\} \quad (5.35)$$

où

$$\lim_{t\to 0} \varepsilon_t' = \lim_{t\to 0} \varepsilon_t'' = 0;$$

ε_t' et ε_t'' ne dépendent pas de X_0 et Y_0.

Compte tenu des inégalités (5.33) et des critères (5.35), on obtient

$$\varepsilon_t' + \min_{Y \in G_Y} \Gamma(X_0, Y) \leqslant \frac{\varphi(t) - \varphi(0)}{t} \leqslant \max_{X \in G_X} \Gamma(X, Y_0) + \varepsilon_t''. \quad (5.36)$$

Rappelons que les inégalités (5.36) sont déduites pour n'importe quels points $X_0 \in G_X = G_X(0)$ et $Y_0 \in G_Y = G_Y(0)$. Donc

$$\varepsilon_t' + \max_{X \in G_X} \min_{Y \in G_Y} \Gamma(X, Y) \leqslant \frac{\varphi(t) - \varphi(0)}{t} \leqslant$$
$$\leqslant \min_{Y \in G_Y} \max_{X \in G_X} \Gamma(X, Y) + \varepsilon_t''. \quad (5.37)$$

D'après le lemme 5.1,

$$\Gamma(X^*, Y^*) = \max_{X \in G_X} \min_{Y \in G_Y} \Gamma(X, Y) = \min_{Y \in G_Y} \max_{X \in G_X} \Gamma(X, Y) = \lambda, \tag{5.38}$$

où (X^*, Y^*) est le point selle de la fonction $\Gamma(X, Y)$ sous la condition $X \in G_X$, $Y \in G_Y$.

(5.37) et (5.38) entraînent le critère

$$\left| \frac{\varphi(t) - \varphi(0)}{t} - \lambda \right| \leqslant \max\{\varepsilon_t', \varepsilon_t''\} = \varepsilon_t, \tag{5.39}$$

où

$$\lim_{t \to 0} \varepsilon_t = 0.$$

En passant dans (5.39) à la limite pour $t \to 0$, on obtient

$$f'_\Delta = \lim_{t \to 0} \frac{\varphi(t) - \varphi(0)}{t} = \lambda. \tag{5.40}$$

Ainsi, l'existence de la dérivée $\varphi'_\Delta(0)$ est établie pour n'importe quelle direction Δ, et les égalités (5.38) et (5.40) sont équivalentes à la relation (5.14). On a ainsi le résultat annoncé par le théorème 5.1.

5.5. Un corollaire du théorème 5.1 nous fournit les formules des dérivées partielles de la fonction $f(a_{ij}, c_j, b_i)$. Désignons par

$$\frac{\partial f}{\partial a_{ij}^+}, \quad \frac{\partial f}{\partial c_j^+}, \quad \frac{\partial f}{\partial b_i^+}$$

les dérivées partielles à droite de la fonction $f(a_{ij}, c_j, b_i)$ par rapport aux arguments correspondants et par

$$\frac{\partial f}{\partial a_{ij}^-}, \quad \frac{\partial f}{\partial c_j^-}, \quad \frac{\partial f}{\partial b_i^-}$$

les dérivées partielles à gauche de cette fonction.

Théorème 5.2. *Sous les hypothèses du théorème* 5.1 :

$$\frac{\partial f}{\partial c_j^+} = \max_{X \in G_X} x_j = x_{j\,max}, \quad \frac{\partial f}{\partial c_j^-} = \min_{X \in G_X} x_j = x_{j\,min}, \tag{5.41}$$

$$\frac{\partial f}{\partial b_i^+} = \min_{Y \in G_Y} y_i = y_{i\,min}, \quad \frac{\partial f}{\partial b_i^-} = \max_{Y \in G_Y} y_i = y_{i\,max}, \tag{5.42}$$

$$\begin{aligned} \frac{\partial f}{\partial a_{ij}^+} &= \begin{cases} -\dfrac{\partial f}{\partial c_j^-} \cdot \dfrac{\partial f}{\partial b_i^-} & si \quad \dfrac{\partial f}{\partial b_i^-} \geqslant 0, \\ -\dfrac{\partial f}{\partial c_j^+} \cdot \dfrac{\partial f}{\partial b_i^-} & si \quad \dfrac{\partial f}{\partial b_i^-} < 0, \end{cases} \\ \frac{\partial f}{\partial a_{ij}^-} &= \begin{cases} -\dfrac{\partial f}{\partial c_j^+} \cdot \dfrac{\partial f}{\partial b_i^+} & si \quad \dfrac{\partial f}{\partial b_i^+} \geqslant 0, \\ -\dfrac{\partial f}{\partial c_j^-} \cdot \dfrac{\partial f}{\partial b_i^+} & si \quad \dfrac{\partial f}{\partial b_i^+} < 0. \end{cases} \end{aligned} \tag{5.43}$$

Démonstration. Soit $\Delta_1^+(j)$ ($\Delta_1^-(j)$) la direction dont toutes les composantes sont nulles, sauf $\delta_{0j} = 1$ ($\delta_{0j} = -1$);

$\Delta_2^+(i)$ ($\Delta_2^-(i)$), la direction dont toutes les composantes, sauf $\delta_{i0} = 1$ ($\delta_{i0} = -1$), sont nulles;

$\Delta^+(ij)$ ($\Delta^-(ij)$), la direction dont l'unique composante non nulle s'écrit

$$\delta_{ij} = 1 \ (\delta_{ij} = -1).$$

Il est évident que

$$\frac{\partial f}{\partial c_j^+} = f'_{\Delta_1^+(j)}, \quad \frac{\partial f}{\partial c_j^-} = -f'_{\Delta_1^-(j)},$$

$$\frac{\partial f}{\partial b_i^+} = f'_{\Delta_2^+(i)}, \quad \frac{\partial f}{\partial b_i^-} = -f'_{\Delta_2^-(i)},$$

$$\frac{\partial f}{\partial a_{ij}^+} = f'_{\Delta^+(ij)}, \quad \frac{\partial f}{\partial a_{ij}^-} = -f'_{\Delta^-(ij)}.$$

Par suite, d'après le théorème 5.1, les dérivées considérées existent toutes; de plus,

$$\frac{\partial f}{\partial c_j^+} = \max_{X \in G_X} x_j, \quad \frac{\partial f}{\partial c_j^-} = -\max_{X \in G_X} (-x_j) = \min_{X \in G_X} x_j,$$

$$\frac{\partial f}{\partial b_i^+} = \min_{Y \in G_Y} y_i, \quad \frac{\partial f}{\partial b_i^-} = -\min_{Y \in G_Y} (-y_i) = \max_{Y \in G_Y} y_i,$$

$$\frac{\partial f}{\partial a_{ij}^+} = \max_{X \in G_X} \min_{Y \in G_Y} (-x_j y_i) = \begin{cases} -x_{j\,min} y_{i\,max} & \text{si} \quad y_{i\,max} \geqslant 0, \\ -x_{j\,max} y_{i\,max} & \text{si} \quad y_{i\,max} < 0, \end{cases}$$

$$\frac{\partial f}{\partial a_{ij}^-} = -\max_{X \in G_X} \min_{Y \in G_Y} (x_j y_i) = \begin{cases} -x_{j\,max} y_{i\,min} & \text{si} \quad y_{i\,min} \geqslant 0, \\ -x_{j\,min} y_{i\,min} & \text{si} \quad y_{i\,min} < 0, \end{cases}$$

ce qu'il fallait démontrer.

5.6. Le théorème principal de ce paragraphe (théorème 5.1) est démontré en partant de l'hypothèse que les ensembles des solutions G_X et G_Y des problèmes A et $\tilde{A}$ sont bornés. La question se pose de savoir s'il n'est pas possible d'étendre la proposition de ce théorème au cas général où G_X et G_Y peuvent être non seulement des polyèdres, mais encore des ensembles convexes.

Nous laissons au lecteur le soin de vérifier (cf. exercice 24) que dans le cas où l'un au moins des ensembles G_X, G_Y est non borné il y a une direction Δ telle que la dérivée f'_Δ n'existe pas. C'est pourquoi l'hypothèse des ensembles G_X et G_Y bornés est essentielle pour justifier toutes les propositions du théorème 5.1.

Signalons que le théorème 5.1 est dû à H. Mills [56], qui a démontré la relation (5.14) pour le cas des jeux rectangulaires. Cependant, la démonstration de ce théorème faite dans [56] pour le problème de programmation linéaire n'est pas suffisamment correcte.

Le théorème 5.1, démontré dans ce paragraphe, transpose aux problèmes de programmation linéaire le résultat obtenu par H. Mills qui n'est rigoureusement justifié que pour les jeux matriciels. ([15] poursuit également ce but).

Si les ensembles G_X et G_Y ne sont pas supposés bornés, (dans [56] on admet l'existence des ensembles G_X et G_Y non bornés), on voit d'après les exemples qu'il est impossible d'assurer l'observation de la relation (5.14) même lorsque cette relation possède une dérivée et un point selle.

Voici un exemple très simple de ce type.

Problème $A(t)$, $t \geqslant 0$:

$$\begin{aligned} & tx_1 \to \max, \\ & x_1 - tx_2 + x_3 = 0, \\ & x_1 + x_4 = 1, \\ & x_1 \geqslant 0,\ x_2 \geqslant 0,\ x_3 \geqslant 0,\ x_4 \geqslant 0. \end{aligned}$$

Problème $\widetilde{A}(t)$, $t \geqslant 0$:

$$\begin{aligned} & y_2 \to \min, \\ & y_1 + y_2 \geqslant t. \\ & -ty_1 \geqslant 0, \\ & y_1 \geqslant 0, \\ & y_2 \geqslant 0. \end{aligned}$$

Il est évident que l'ensemble $G_X(t)$ des solutions du problème $A(t)$ est défini par la relation

$$G_X(t) = \begin{cases} x_1 = 1,\ x_2 \geqslant \frac{1}{t} & \text{si} \quad t > 0, \\ x_1 = 0,\ x_2 \geqslant 0 & \text{si} \quad t = 0, \end{cases}$$

alors que l'ensemble $G_Y(t)$ est imposé par les conditions

$$G_Y(t) = \begin{cases} y_1 = 0, & y_2 = t \quad \text{si} \quad t > 0, \\ y_1 \geqslant 0, & y_2 = 0 \quad \text{si} \quad t = 0. \end{cases}$$

Pour l'exemple considéré, la fonction $\varphi(t) = t$, $t \geqslant 0$. Par conséquent,

$$f'_\Delta = \frac{d\varphi(t)}{dt}\Big|_{t=0} = 1,$$

où la direction $\Delta = \{\delta_{ij}, \delta_{0j}, \delta_{i0}\}$ se construit d'après les coefficients de t dans les conditions du problème A:

$$\delta_{ij} = \begin{cases} 1 \text{ si } i = 0,\ j = 1, \\ -1 \text{ si } i = 1,\ j = 2, \\ 0 \text{ dans les autres cas.} \end{cases}$$

Pour l'exemple considéré, la fonction $\Gamma(X, Y)$ s'écrit

$$\Gamma(X, Y) = x_1 + y_1 x_2.$$

Assurons-nous que $\Gamma(X, Y)$ a un point selle sous la condition $(X, Y) \in G_X \times G_Y$. En effet,

$$\max_{X \in G_X} \min_{Y \in G_Y} \Gamma(X, Y) = \max_{x_1=0,\ x_2 \geqslant 0,\ y_1 \geqslant 0} \min [x_1 + y_1 x_2] =$$

$$= \max_{x_2 \geqslant 0} \min_{y_1 \geqslant 0} (y_1 x_2) = \min_{y_1 \geqslant 0} \max_{x_2 \geqslant 0} (y_1 x_2) = 0 = \Gamma(0, 0).$$

Ainsi, dans l'exemple considéré la dérivée f'_Δ et le point selle (X^*, Y^*) de la fonction $\Gamma(X, Y)$ existent, et cependant

$$f'_\Delta = 1 \neq \Gamma(X^*, Y^*) = 0,$$

c'est-à-dire que la relation (5.14) n'est pas vérifiée.

Nous laissons au lecteur le soin de construire, en partant de l'exemple donné, un autre exemple dans lequel la fonction $\varphi(t)$ est définie pour $t \geqslant 0$, la fonction $\Gamma(X, Y)$ admet un point selle et néanmoins la dérivée f'_Δ n'existe pas (exercice 25).

5.7. A titre de conclusion nous allons démontrer une proposition qui précise le théorème 5.1 pour une certaine classe de directions Δ. Associons dans l'ensemble H toutes les directions $\Delta = \{\delta_{ij}, \delta_{0j}, \delta_{i0}\}$, telles que leurs composantes $\delta_{ij} = 0$ avec $i = 1, 2, \ldots, m$; $j = 1, 2, \ldots, n$. Si $\Delta \in H$,

$$\Gamma(X, Y) = \sum_{j=1}^{n} \delta_{0j} x_j + \sum_{i=1}^{m} \delta_{i0} y_i.$$

En supposant que la première somme soit bornée supérieurement sur l'ensemble G_X et que la deuxième soit bornée inférieurement sur l'ensemble G_Y, on a

$$\min_{Y \in G_Y} \max_{X \in G_X} \Gamma(X, Y) = \max_{X \in G_X} \min_{Y \in G_Y} \Gamma(X, Y) =$$

$$= \max_{X \in G_X} \Big(\sum_{j=1}^{n} \delta_{0j} x_j\Big) + \min_{Y \in G_Y} \Big(\sum_{i=1}^{m} \delta_{i0} y_i\Big).$$

Théorème 5.3. *Soit le problème $A = A(0)$ à solution réalisable. Si $\Delta \in H$, l'existence de la dérivée f'_Δ est équivalente à la vérification des conditions*

$$\sup_{X \in G_X} \Big(\sum_{j=1}^{n} \delta_{0j} x_j\Big) < \infty, \quad \inf_{Y \in G_Y} \Big(\sum_{i=1}^{m} \delta_{i0} y_i\Big) > -\infty, \tag{5.44}$$

ou, ce qui revient au même, à la présence d'un point selle dans la fonction $\Gamma(X, Y)$. Lorsque les conditions (5.44) sont respectées,

$$f'_\Delta = \max_{X \in G_X} \Big(\sum_{j=1}^{n} \delta_{0j} x_j\Big) + \min_{Y \in G_Y} \Big(\sum_{i=1}^{m} \delta_{i0} y_i\Big). \tag{5.45}$$

Démonstration. Appliquons au problème $A(t)$, $t \geqslant 0$, l'algorithme de programmation paramétrique exposé aux paragraphes 3.2, 3.3. Si l'on prend comme programme initial une solution de base arbitraire du problème $A = A(0)$ et si l'on tâche, au cours de la discussion du problème paramétrique, d'accroître la valeur du paramètre, on obtient en un nombre fini d'étapes des méthodes primale et duale du simplexe une base B du problème A pour laquelle se réalise l'une des trois éventualités:

a) $x_{i0}^{(1)} \geqslant 0, \quad x_{i0}^{(2)} \geqslant 0 \quad \text{si} \quad x_{i0}^{(1)} = 0$

$\Delta_j^{(1)} \geqslant 0, \quad \Delta_j^{(2)} \geqslant 0 \quad \text{si} \quad \Delta_j^{(1)} = 0;$

b) pour un certain k

$$\Delta_k^{(2)} < 0, \quad \Delta_k^{(1)} = 0, \quad x_{ik} \leqslant 0, \quad i = 1, 2, \ldots, m;$$

c) pour un certain r

$$x_{r0}^{(2)} < 0, \quad x_{r0}^{(1)} = 0, \quad x_{rj} \geqslant 0, \quad j = 1, 2, \ldots, n.$$

Les x_{ij} sont ici les coefficients de décomposition des vecteurs des conditions A_j du problème A par rapport à la base B;

$x_{i0}^{(1)}$ et $x_{i0}^{(2)}$, les coefficients de décomposition du vecteur V et du vecteur $(\delta_{10}, \delta_{20}, \ldots, \delta_{m0})^T$ par rapport à la base B;

$\Delta_j^{(1)}$ et $\Delta_j^{(2)}$, les estimations des vecteurs A_j par rapport à B avec les coefficients de la forme linéaire c_j et δ_{0j} $(j = 1, 2, \ldots, n)$ respectivement.

Soit, pour fixer les idées, $B = (A_1, A_2, \ldots, A_m)$.

a) Si l'on est dans le cas a), le programme de base $X(t)$ aux composantes de base

$$x_i(t) = x_{i0}^{(1)} + t x_{i0}^{(2)}, \quad i = 1, 2, \ldots, m,$$

est solution du problème $A(t)$ pour $t \in [0, t_0]$, où t_0 est un certain nombre positif.

C'est pourquoi, avec $t \in [0, t_0]$,

$$\varphi(t) = f(a_{ij}, c_j + t\delta_{0j}, b_i + t\delta_{i0}) = \sum_{i=1}^{m} (c_i + t\delta_{0i})(x_{i0}^{(1)} + t x_{i0}^{(2)})$$

et, par conséquent,

$$f'_\Delta = \frac{d\varphi}{dt}\Big|_{t=0} = \sum_{i=1}^{m} \delta_{0i} x_{i0}^{(1)} + \sum_{i=1}^{m} c_i x_{i0}^{(2)}. \tag{5.46}$$

Soit

$$\Delta_j^{(1)} = \sum_{i=1}^{m} a_{ij} y_i^{(1)} - c_j, \quad \Delta_j^{(2)} = \sum_{i=1}^{m} a_{ij} y_i^{(2)} - \delta_{0j}.$$

Nous sommes en présence de la suite d'égalités

$$\sum_{i=1}^{m} c_i x_{i0}^{(2)} = \sum_{j=1}^{m} \Big(\sum_{i=1}^{m} a_{ij} y_i^{(1)} \Big) x_{j0}^{(2)} = \sum_{i=0}^{m} y_i^{(1)} \Big(\sum_{j=1}^{m} a_{ij} x_{j0}^{(2)} \Big) = \sum_{i=1}^{m} y_i^{(1)} \delta_{i0},$$

qui permet de mettre la relation (5.46) sous la forme

$$f'_\Delta = \sum_{i=1}^{m} \delta_{0i} x_{i0}^{(1)} + \sum_{i=1}^{m} \delta_{i0} y_i^{(1)}. \tag{5.47}$$

Le système de vecteurs B est une base optimale du problème $A = A\ (0)$. C'est pourquoi le programme de base X_1 du problème A aux composantes de base $x_{i0}^{(1)}$, $i = 1, 2, \ldots, m$, constitue une solution de ce problème, et le vecteur $Y_1 = (y_1^{(1)}, y_2^{(1)}, \ldots, y_m^{(1)})$, un programme optimal du problème A.

Soit X une solution quelconque du problème A. D'après le deuxième théorème de dualité (§ 2, chapitre 6 de [87]), $x_j = 0$ si $\Delta_j^{(1)} > 0$.

Par conséquent,

$$\sum_{j=1}^{n} \delta_{0j} x_j = \sum_{j=1}^{n} \delta_{0j} x_j^{(1)} - \sum_{j=1}^{n} \Delta_j^{(2)} x_j =$$

$$= \sum_{j=1}^{n} \delta_{0j} x_j^{(1)} - \sum_{\Delta_j^{(1)}=0} \Delta_j^{(2)} x_j \leqslant \sum_{j=1}^{n} \delta_{0j} x_j^{(1)},$$

étant donné que $\Delta_j^{(2)} \geqslant 0$ si $\Delta_j^{(1)} = 0$. On obtient ainsi la relation

$$\sum_{j=1}^{n} \delta_{0j} x_j^{(1)} = \sum_{i=1}^{m} \delta_{0i} x_{i0}^{(1)} = \max_{X \in G_X} \sum_{j=1}^{n} \delta_{0j} x_j < \infty. \tag{5.48}$$

Pour vérifier la relation

$$\sum_{i=1}^{m} \delta_{i0} y_i^{(1)} = \min_{Y \in G_Y} \sum_{i=1}^{m} \delta_{i0} y_i > -\infty, \tag{5.49}$$

on procède d'une façon analogue.

Ainsi, l'éventualité a) assure l'existence de la dérivée f'_Δ, et de plus, donne lieu à la relation (5.45).

b) Supposons que ce soit l'éventualité b) qui est vérifiée pour la base $B = (A_1, A_2, \ldots, A_m)$. Dans ce cas, le problème $\tilde{A}'(t)$ n'a pas de programmes avec $t > 0$, ce qui signifie que la fonction $\varphi(t)$ n'est pas définie à droite du point $t = 0$. En effet, soient $t > 0$ et Y le vecteur quelconque vérifiant les inégalités

$$(A_i, Y) \geqslant c_i + \delta_{i0} t, \quad i = 1, 2, \ldots, m.$$

Compte tenu des coefficients x_{ik} non positifs, on a

$$(A_k,\ Y)=\sum_{i=1}^{m} x_{ik}(A_i,\ Y)\leqslant\sum_{i=1}^{m}(c_i+\delta_{0i}t)\,x_{ik}=$$
$$=\Delta_k^{(1)}+t\Delta_k^{(2)}+c_k+t\delta_{0k}=t\Delta_k^{(2)}+c_k+t\delta_{0k}<c_k+t\delta_{0k},$$

puisque $\Delta_k^{(1)}=0$, $\Delta_k^{(2)}<0$, $t>0$. Ainsi, il n'existe pas de vecteur vérifiant toutes les conditions du problème $\widetilde{A}\ (t)$ avec $t>0$, ce qu'il fallait démontrer.

D'après les conditions du cas b), le vecteur $X\ (\theta)=X_1+\theta z$, où

$$z_j=\begin{cases}-x_{jk} & \text{avec} \quad j=1,\ 2,\ \ldots,\ m,\\ \ \ 1 & \text{avec} \quad j=k,\\ \ \ 0 & \text{dans les autres cas,}\end{cases}$$

est solution du problème A pour tout $\theta\geqslant 0$.

Par ailleurs,

$$\lim_{\theta\to\infty}\sum_{j=1}^{n}\delta_{0j}x_j(\theta)=\lim_{\theta\to\infty}\Big[\sum_{j=1}^{n}\delta_{0j}x_j^{(1)}-\theta\Delta_k^{(2)}\Big]=\infty,$$

c'est-à-dire

$$\sup_{X\in G_X}\sum_{j=1}^{n}\delta_{0j}x_j=\infty. \tag{5.50}$$

c) Considérons enfin la dernière éventualité. Si c'est le cas c) qui a lieu, le problème $A\ (t)$ n'a pas de programmes pour $t>0$, ce qui signifie que la fonction $\varphi\ (t)$ n'est pas définie pour les valeurs positives de t. En effet, avec $t>0$ et un vecteur X quelconque aux composantes x_j non négatives, l'égalité

$$x_{r0}^{(1)}+tx_{r0}^{(2)}=\sum_{j=1}^{n}x_{rj}x_j$$

est impossible, puisque

$$x_{r0}^{(1)}+tx_{r0}^{(2)}=tx_{r0}^{(2)}<0,$$

alors que

$$\sum_{j=1}^{n}x_{rj}x_j\geqslant 0.$$

Notons que le vecteur $Y\ (\theta)=Y_1+\theta e_r$, où e_r est la r-ième ligne de la matrice inverse à celle de la base, constitue un programme optimal du problème $\widetilde{A}$ quel que soit $\theta\geqslant 0$. En outre,

$$\lim_{\theta\to\infty}\sum_{i=1}^{m}\delta_{i0}y_i(\theta)=\lim_{\theta\to\infty}\Big[\sum_{i=1}^{m}\delta_{i0}y_i^{(1)}+x_{r0}^{(2)}\theta\Big]=-\infty,$$

c'est-à-dire

$$\inf_{Y \in G_Y} \sum_{i=1}^{m} \delta_{i0} y_i = -\infty. \tag{5.51}$$

Il n'est plus difficile maintenant d'achever la démonstration du théorème. Supposons que les conditions (5.44) soient vérifiées. Alors (5.50) et (5.51) rendent impossible la réalisation des cas b) et c). Nous sommes donc ramenés au cas a), et, comme on l'a montré, la relation (5.45) est respectée.

Mais si les conditions (5.44) ne sont pas observées, le cas a) est impossible [cf. (5.48) et (5.49)], et donc c'est le cas b) ou le cas c) qui se réalisent. Comme il a été montré, dans chacun de ces cas la fonction $\varphi(t)$ n'est pas définie pour des t positifs et par suite, la dérivée f'_Δ n'existe pas. Le théorème 5.3 est ainsi démontré.

Constatons que la proposition du théorème 5.3 est vraie notamment pour les dérivées $\frac{\partial f}{\partial c_j^+}$, $\frac{\partial f}{\partial b_i^+}$, $\frac{\partial f}{\partial c_j^-}$, et $\frac{\partial f}{\partial b_i^-}$. Toutefois son extension aux dérivées $\frac{\partial f}{\partial a_{ij}^+}$ et $\frac{\partial f}{\partial a_{ij}^-}$ ne semble pas possible. Nous proposons au lecteur de s'en convaincre par lui-même en construisant un exemple correspondant (cf. exercice 26).

Exercices

1. Montrer que chacune des éventualités indiquées par le théorème 1.1 est possible.
2. Démontrer que l'algorithme de programmation paramétrique du § 1 est fini sous la condition que la procédure de calcul du problème (1.1)-(1.3) avec $\lambda = \lambda_0$ aboutisse au cas 2°.
3. Explorer le problème

$$\begin{aligned} L(X) = (1-2\lambda)x_1 + \lambda x_2 + x_3 - (2-\lambda)x_4 &\to \max, \\ 3x_1 - x_2 + 2x_3 + 5x_4 &\leqslant 10, \\ x_1 + 3x_3 + x_4 &\leqslant 6, \\ 2x_1 - 3x_2 - x_3 &\leqslant 5, \\ x_j \geqslant 0,\ j = 1, 2, 3, 4, \end{aligned}$$

pour toutes les valeurs du paramètre λ.

4. Montrer que la fonction $f(\lambda)$ du théorème 1.3 est convexe vers le bas.
5. Montrer que chacune des éventualités indiquées par le théorème 2.1 est possible.
6. Démontrer que l'algorithme de programmation linéaire du § 2 est fini.
7. Discuter le problème

$$\begin{aligned} L(X) = 2x_1 - x_2 + 5x_3 + x_4 &\to \max, \\ x_1 + 2x_2 + x_3 + 3x_4 &\leqslant 10 + 3\mu, \\ 4x_1 - x_2 + 2x_3 + x_4 &\leqslant 7 - \mu, \\ 3x_2 + x_3 - 2x_4 &\leqslant 2 + 5\mu, \\ x_j \geqslant 0,\ j = 1, 2, 3, 4 \end{aligned}$$

pour toutes les valeurs du paramètre μ.

8. Démontrer le théorème 2.3.

9. Analyser le problème

$$L(X) = (-1+t)\,x_1 + (-2+t)\,x_2 + (-3+2t)\,x_3 + (-1-2t)\,x_4 \to \max,$$

$$\begin{aligned} x_1 - 2x_2 + x_3 - x_4 &\leqslant 1+t, \\ 2x_1 + 3x_2 - x_3 + 2x_4 &\leqslant 2-3t, \\ -x_1 + 2x_2 + 3x_3 - 3x_4 &\leqslant 3-2t, \\ x_j \geqslant 0,\ j &= 1, 2, 3, 4, \end{aligned}$$

pour toutes les valeurs du paramètre t.

10. Démontrer le théorème 3.1.

11. Montrer que l'algorithme de programmation paramétrique décrit aux paragraphes 3.2, 3.3 est fini.

12. Démontrer le théorème 3.3.

13. Justifier la proposition du théorème 3.4.

14. Explorer le problème

$$\begin{aligned} L(X) = 2x_1 + x_2 + 4x_3 + 3x_4 &\to \max, \\ (2+t)\,x_1 + x_2 + 2x_3 + 5x_4 &\leqslant 12, \\ (1-2t)\,x_1 - 2x_2 \qquad + 3x_4 &\leqslant 7, \\ (3+4t)\,x_1 + 3x_2 - x_3 + 2x_4 &\leqslant 9, \\ x_j \geqslant 0,\ j &= 1, 2, 3, 4, \end{aligned}$$

pour toutes les valeurs du paramètre t.

15. Appliquer le schéma du paragraphe 3.6 pour étudier le problème

$$\begin{aligned} (3+t)\,x_1 - x_2 + tx_3 &\to \max, \\ (2+\operatorname{ch} t)\,x_1 + (2-\operatorname{sh} t)\,x_2 + x_3 &\leqslant 2-t, \\ (2-\operatorname{sh} t)\,x_1 - (2+\operatorname{ch} t)\,x_2 \qquad &\leqslant 15+4t, \\ x_j \geqslant 0,\ j &= 1, 2, 3, \end{aligned}$$

pour toutes les valeurs du paramètre t.

16. Composer un schéma de résolution du problème paramétrique où les éléments d'une des lignes de la matrice des conditions dépendent linéairement du paramètre.

17. Donner un exemple de problème du type (3.13)-(3.15) dont la valeur maximale de la forme linéaire est une fonction discontinue du paramètre t.

18. Etendre l'algorithme de résolution du problème (4.1)-(4.4) au cas d'un ensemble convexe quelconque M défini par les conditions (4.2), (4.3).

19. Justifier l'algorithme de résolution du problème (4.27)-(4.29) donné au paragraphe 4.4.

20. Décrire l'algorithme de résolution du problème (4.27)-(4.29) sans supposer que les conditions (4.29) et (4.30) sont compatibles pour un μ quelconque.

21. Résoudre le problème de programmation homographique

$$R(X) = \frac{\sum_{i=1}^{3}\sum_{j=1}^{4} c_{ij}x_{ij}}{\sum_{i=1}^{3}\sum_{j=1}^{4} d_{ij}x_{ij} + d} \to \max,$$

$$\sum_{j=1}^{4} x_{ij} = a_i, \quad i = 1, 2, 3;$$

$$\sum_{i=1}^{3} x_{ij} = b_j, \quad j = 1, 2, 3, 4;$$

$$x_{ij} \geqslant 0, \quad i = 1, 2, 3; \quad j = 1, 2, 3, 4,$$

où

$$\|c_{ij}\| = \begin{Vmatrix} 3 & 5 & 2 & 8 \\ 7 & 4 & 10 & 3 \\ 6 & 8 & 5 & 7 \end{Vmatrix}; \quad \|d_{ij}\| = \begin{Vmatrix} 4 & 1 & 3 & 2 \\ 0 & 5 & 0 & 4 \\ 3 & 0 & 6 & 0 \end{Vmatrix};$$

$(a_1, a_2, a_3) = (10, 20, 25)$; $(b_1, b_2, b_3, b_4) = (12, 6, 19, 18)$; $d = 1$.

22. Composer un algorithme de résolution du problème (4.40)-(4.42) muni d'une contrainte supplémentaire sur le dénominateur de la fraction $R(X)$ du type

$$\beta_1 \leqslant L_2(X) \leqslant \beta_2.$$

23. Démontrer que si le problème (4.40)-(4.42) comporte un maximum local, différent du maximum global, la face supérieure de la fraction $R(X)$ sous les conditions (4.41), (4.42) est égale à l'infini. Donner un exemple de problème du type (4.40)-(4.42) ayant un maximum local différent du maximum global.

24. Montrer que si l'un au moins des ensembles G_X ou G_Y est non borné pour une certaine direction Δ la fonction $\varphi(t)$ n'est pas définie avec $t > 0$, c'est-à-dire la dérivée f'_Δ n'existe pas.

25. Construire un problème de programmation linéaire et une direction Δ tels que la fonction $\varphi(t)$ soit définie pour $t \geqslant 0$, la fonction $\Gamma(X, Y)$ ait un point selle et la dérivée f'_Δ n'existe cependant pas.

26. Montrer que la proposition du théorème 5.3 ne peut pas être étendue aux dérivées $\dfrac{\partial f}{\partial a_{ij}^+}$ et $\dfrac{\partial f}{\partial a_{ij}^-}$.

27. Soit $\varphi(t)$ la valeur maximale de la forme linéaire

$$\sum_{j=1}^{n} c_j(t)\, x_j$$

sous les conditions

$$\sum_{j=1}^{n} a_{ij}(t)\, x_j = b_i(t), \quad i = 1, 2, \ldots, m,$$

$$x_j \geqslant 0, \quad j = 1, 2, \ldots, n,$$

et $A(t)$ le problème définissant la fonction $\varphi(t)$.

Montrer que si les ensembles des solutions $G_X(t_0)$ et $G_Y(t_0)$ des problèmes $A(t_0)$ et $\widetilde{A}(t_0)$ sont bornés, et si chacune des fonctions $c_j(t)$, $b_i(t)$, $a_{ij}(t)$ a une dérivée à droite en t_0, il existe une dérivée à droite $\dfrac{d\varphi(t_0)}{dt^+}$ de la fonction $\varphi(t)$ n $t = t_0$, et de plus

$$\frac{d\varphi(t_0)}{dt^+} = \Gamma(X^*, Y^*),$$

où (X^*, Y^*) est un point selle de la fonction $\Gamma(X, Y)$ sous les conditions $(X, Y) \in G_X(t_0) \times G_Y(t_0)$;

$$\Gamma(X, Y) = \sum_{i=1}^{m} \frac{db_i(t_0)}{dt^+} y_i + \sum_{j=1}^{n} \frac{dc_j(t_0)}{dt^+} x_j -$$

$$- \sum_{i=1}^{m} \sum_{j=1}^{n} \frac{da_{ij}(t_0)}{dt^+} y_i x_j.$$

28. Supposons que les conditions (4.41), (4.42) assurent soit la non-négativité, soit la non-positivité de la fonction $L_2(X)$. En utilisant le théorème 4.1, démontrer la proposition suivante (Schvartzman A. P.).

Un programme de base X^* ($L_2(X^*) \neq 0$) à base B est solution du problème (4.40)-(4.42) si

$$\begin{vmatrix} \Delta_j^{(1)} & L_1(X^*) \\ \Delta_j^{(2)} & L_2(X^*) \end{vmatrix} \geqslant 0, \quad j = 1, 2 \ldots, n.$$

où $\Delta_j^{(1)}$ et $\Delta_j^{(2)}$ sont les critères de sélection du j-ième vecteur des conditions par rapport à la base B aux coefficients de la forme linéaire définis par $L_1(X)$ et $L_2(X)$ respectivement.

Chapitre 4

PROGRAMMATION PAR BLOCS

Le nombre de contraintes intervenant dans les problèmes pratiques de programmation linéaire se mesure par centaines et parfois par milliers d'égalités et d'inégalités. Le nombre de variables dans les problèmes extrémaux économiques et techniques complexes se chiffre par milliers. Le temps nécessaire pour résoudre ces problèmes est déterminé principalement par le nombre d'appels à la mémoire centrale de l'ordinateur. Lors de la résolution de problèmes extrémaux conditionnels de grande taille, la rapidité effective de la machine peut être grandement augmentée si l'on modifie les schémas de calcul de façon à diminuer le nombre d'appels à la mémoire centrale au prix d'un certain accroissement du nombre d'opérations élémentaires. De là vient que l'on s'efforce de ramener la résolution de problèmes à grand nombre de variables et de contraintes à celle de quelques problèmes à plus petit nombre de variables et de contraintes.

L'ensemble des procédés qui ramènent la résolution des problèmes de grande taille à une exploration de quelques problèmes de taille moindre s'appelle *programmation par blocs.*

L'avantage de cette discipline réside non seulement dans l'utilisation rationnelle de la mémoire à accès immédiat de la machine. La programmation par blocs permet souvent de diviser un problème dont la structure de la matrice des conditions interdit l'utilisation de méthodes spéciales de résolution, en quelques problèmes de taille plus faible dont les conditions autorisent l'application de méthodes spéciales.

L'un des principes possibles de programmation par blocs est proposé par [38] et [39]. Il a reçu le nom de *méthode de décomposition.* La description de cette méthode et de ses modifications possibles fait l'objet des §§ 1 et 2. Cette méthode peut être considérée comme l'analogue par blocs de la méthode primale du simplexe.

Les §§ 3 à 8 donnent une approche assez générale d'élaboration de méthodes de programmation par blocs, qui peut servir notamment pour l'obtention de la méthode de décomposition.

Le § 3 traite de l'attitude à prendre devant la programmation par blocs et de son interprétation géométrique. Le § 4 est consacré à la description de l'analogue dual de la méthode de décomposition, qui est fondé sur la méthode duale du simplexe. Dans le § 5 on examine les méthodes de programmation par blocs, dont le principe les apparente à la méthode primale duale.

Dans les §§ 6 et 7, on discute les moyens de construction de l'approximation initiale et l'on justifie la convergence de ces méthodes. La méthode de programmation par blocs proche de la méthode d'encadrement est décrite au § 8.

Ainsi les premiers huit paragraphes du chapitre comprennent les analogues par blocs de toutes les méthodes finies principales de programmation linéaire.

Les méthodes de programmation par blocs peuvent être utilisées pour la construction d'algorithmes efficaces de résolution des problèmes spéciaux de programmation linéaire. Les §§ 9 et 10 illustrent les possibilités des méthodes par blocs sur les exemples du problème de transport, du problème de distribution et du problème à trois indices à sommes axiales.

Les conceptions de la programmation par blocs ne font pas que simplifier la procédure de calcul, mais étendent aussi dans bien des cas le domaine d'application de la programmation linéaire. Dans le § 11, les possibilités de la programmation par blocs sont illustrées par l'exemple d'un problème linéaire à vecteurs des conditions non fixés d'avance.

§ 1. Méthode de décomposition

1.1. Considérons le problème de programmation linéaire de type général qui consiste à maximiser la forme linéaire

$$L = (C, X) \tag{1.1}$$

sous les conditions

$$AX = B, \tag{1.2}$$

$$X \geqslant 0. \tag{1.3}$$

$C = (c_1, c_2, \ldots, c_n)$ est ici le vecteur-ligne de dimension n;

$B = (b_1, b_2, \ldots, b_{m+m_1})^T$, le vecteur-colonne de dimension $m + m_1$;

$A = \| a_{ij} \|$, la matrice des conditions de dimensions $(m + m_1) \times \times n$;

$X = (x_1, x_2, \ldots, x_n)^T$, le vecteur-colonne de dimension n.

Divisons les contraintes du problème en deux blocs dont le premier comprend les m premières contraintes, et le deuxième les contraintes restantes. Désignons par $A^{(0)}$ et $A^{(1)}$ respectivement les matrices des conditions associées aux deux blocs, et par $B^{(0)}$

et $B^{(1)}$ les vecteurs des contraintes des blocs.

$$A=\begin{Vmatrix} A^{(0)} \\ A^{(1)} \end{Vmatrix}; \quad B=\begin{pmatrix} B^{(0)} \\ B^{(1)} \end{pmatrix}.$$

Le problème (1.1)-(1.3) s'écrit alors

$$L = (C,\ X) \to \max \tag{1.4}$$

sous les conditions

$$A^{(0)}\ X = B^{(0)}, \tag{1.5}$$

$$A^{(1)}\ X = B^{(1)}, \tag{1.6}$$

$$X \geqslant 0. \tag{1.7}$$

Supposons que l'ensemble défini par les conditions (1.6), (1.7) soit borné (par la suite, nous renoncerons à cette hypothèse). Désignons par $X_1,\ X_2,\ \ldots,\ X_N$ les sommets du polyèdre convexe répondant aux conditions (1.6), (1.7). Les points $X_1,\ X_2,\ \ldots,\ X_N$ sont les images géométriques des programmes de base du problème (1.4), (1.6), (1.7). Appelons le problème (1.4), (1.6), (1.7) $X^{(1)}$ -problème et le problème initial (1.4)-(1.7), X-problème.

Tout point X du polyèdre (1.6), (1.7) est une combinaison convexe de ses sommets, soit

$$X = \sum_{\nu=1}^{N} z_\nu X_\nu, \tag{1.8}$$

où

$$\sum_{\nu=1}^{N} z_\nu = 1\,; \tag{1.9}$$

$$z\nu \geqslant 0, \quad \nu = 1,\ 2,\ \ldots,\ N. \tag{1.10}$$

Dans le problème initial les formules (1.8)-(1.10) permettent de passer des variables x_j aux variables z_ν. Les programmes de base $X_\nu = (x_1^{(\nu)},\ x_2^{(\nu)},\ \ldots,\ x_n^{(\nu)})$ étant supposés connus, mettons le problème (1.1)-(1.3) [où, ce qui revient au même, le problème (1.4)-(1.7)] sous la forme

$$\sum_{\nu=1}^{N} (C,\ X_\nu)\, z_\nu \to \max$$

sous les conditions:

$$\sum_{\nu=1}^{N} (A^{(0)} X_\nu)\, z_\nu = B^{(0)},$$

$$\sum_{\nu=1}^{N} z_\nu = 1,$$

$$z_\nu \geqslant 0, \quad \nu = 1,\ 2,\ \ldots,\ N.$$

Introduisons les notations

$$(C,\ X_\nu) = \sigma_\nu, \tag{1.11}$$

$$A^{(0)}\, X_\nu = P_\nu. \tag{1.12}$$

Les σ_ν sont ici des scalaires;
les P_ν, des vecteurs de dimension m.

Dans les nouvelles notations, le problème se ramène au calcul des variables z_ν qui maximisent la forme linéaire

$$\sum_{\nu=1}^{N} \sigma_\nu z_\nu \tag{1.13}$$

sous les conditions

$$\sum_{\nu=1}^{N} P_\nu z_\nu = B^{(0)}, \tag{1.14}$$

$$\sum_{\nu=1}^{N} z_\nu = 1, \tag{1.15}$$

$$z \geqslant 0, \quad \nu = 1,\ 2,\ \ldots,\ N. \tag{1.16}$$

Appelons le problème (1.13)-(1.16) Z-problème.

L'écriture des conditions (1.14) et (1.15) du Z-problème devient très commode si on les met sous la forme d'une égalité vectorielle

$$\sum_{\nu=1}^{N} \overline{P}_\nu z_\nu = \overline{B}^{(0)}, \tag{1.17}$$

où

$$\left.\begin{aligned} \overline{P}_\nu &= \begin{pmatrix} P_\nu \\ 1 \end{pmatrix}, \quad \nu = 1,\ 2,\ \ldots,\ N; \\ \overline{B}^{(0)} &= \begin{pmatrix} B^{(0)} \\ 1 \end{pmatrix} \end{aligned}\right\} \tag{1.18}$$

sont respectivement les vecteurs des conditions et le vecteur des contraintes du Z-problème.

Pour résoudre le Z-problème, il n'est nullement nécessaire de connaître tous les vecteurs $\overline{P}_\nu$. Il faut seulement avoir à sa disposition à chaque étape $m + 1$ vecteurs des conditions du Z-problème qui constituent sa base courante. Quant au vecteur à inclure dans la base, son choix est guidé, comme nous allons le voir, par la résolution d'un problème auxiliaire de programmation linéaire soumis aux conditions (1.6), (1.7). C'est en cela que réside la nature même de la méthode de décomposition.

La résolution du Z-problème détermine univoquement le programme optimal du X-problème. Le vecteur X^*, solution du X-problème, se calcule à l'aide des composantes z_ν^* du programme optimal

du Z-problème d'après la formule

$$X^* = \sum_{\nu=1}^{N} z_\nu^* X_\nu. \tag{1.19}$$

En effet, le vecteur X^* est un programme du X-problème, puisque :

1) X^* vérifie les conditions (1.6), (1.7) en tant que combinaison convexe des programmes de base X_ν ;

2) X^* vérifie les conditions (1.5), car, en vertu de (1.19), (1.12) et (1.14),

$$A^{(0)}X^* = \sum_{\nu=1}^{N} A^{(0)}X_\nu z_\nu^* = \sum_{\nu=1}^{N} P_\nu z_\nu^* = B^{(0)}.$$

En outre, la condition d'optimalité du vecteur Z^*

$$\sum_{\nu=1}^{N} \sigma_\nu z_\nu^* \geqslant \sum_{\nu=1}^{N} \sigma_\nu z_\nu$$

(pour tous les programmes du Z-problème) entraîne

$$\sum_{\nu=1}^{N} (C,\ X_\nu)\, z_\nu^* \geqslant \sum_{\nu=1}^{N} (C,\ X_\nu)\, z_\nu.$$

En vertu de (1.19) et (1.8), cette dernière inégalité peut se mettre sous la forme

$$CX^* \geqslant CX$$

pour tous les programmes du X-problème.

La résolution du X-problème se ramène donc au calcul d'un programme optimal du Z-problème.

Les dimensions du X-problème sont égales à $(m + m_1) \times n$. Le Z-problème compte moins de contraintes $(m + 1)$ mais beaucoup plus de variables (N). Le nombre N de sommets du polyèdre (1.6), (1.7), pour des dimensions assez grandes du deuxième bloc, est un nombre astronomique.

1.2. Considérons le schéma général de résolution du Z-problème. Cherchons sa solution en utilisant le deuxième algorithme primal du simplexe.

Supposons connus les vecteurs de la base d'un certain programme de base du Z-problème (pour la constructon d'une base initiale, cf. paragraphe 1.4). Conformément aux règles de la méthode primale du simplexe, l'ordre des calculs est défini par les valeurs Δ_ν des critères de sélection des vecteurs des conditions par rapport à la base courante. Lorsqu'on utilise le second algorithme, les critères se calculent d'après la formule

$$\Delta_\nu = (\overline{P}_\nu,\ \overline{\Lambda}) - \sigma_\nu, \nu = 1,\ 2,\ \ldots,\ N. \tag{1.20}$$

Le vecteur $\overline{\Lambda}$ des critères relatifs des conditions du Z-problème est de plus une solution du système d'équations

$$(\overline{P}_j, \overline{\Lambda}) = \sigma_j, \tag{1.21}$$

où l'indice j parcourt les numéros des vecteurs de la base. Le programme de base associé à la base considérée est optimal si tous les Δ_ν sont non négatifs. Toutefois, cette forme du critère d'optimalité d'un programme est inefficace pour le Z-problème, puisqu'elle suppose connus tous les vecteurs P_ν, dont le nombre est généralement très grand. Dantzig et Wolfe ont indiqué (et c'est là que réside le fondement de la méthode de décomposition) un moyen pratiquement valable pour tester l'optimalité d'un programme de base du Z-problème et choisir le vecteur à ajouter à la base si le programme exploré n'est pas optimal. Voici le procédé en question.

Prélevons la dernière ($m + 1^{\text{ième}}$) composante du vecteur $\overline{\Lambda}$, associée à la condition (1.15) du Z-problème et désignons-la par λ. Le vecteur $\overline{\Lambda}$ peut alors s'écrire

$$\overline{\Lambda} = (\Lambda, \lambda). \tag{1.22}$$

Les composantes du vecteur Λ sont les critères relatifs des conditions (1.14) du Z-problème $[\Lambda = (\lambda_1, \lambda_2, \ldots, \lambda_m)]$.

Compte tenu des notations (1.18) et (1.22), réécrivons les formules (1.21) et (1.20) des critères relatifs des conditions du problème et des vecteurs des conditions:

$$(\Lambda, P_j) + \lambda = \sigma_j, \quad j \in I_z, \tag{1.23}$$

$$\Delta_\nu = (\Lambda, P_\nu) + \lambda - \sigma_\nu, \quad \nu = 1, 2, \ldots, N, \tag{1.24}$$

où I_z est l'ensemble des indices des vecteurs de la base courante du Z-problème.

Introduisons dans la discussion le vecteur

$$C_\Lambda = C - \Lambda A^{(0)}. \tag{1.25}$$

Les composantes $c_j^{(\Lambda)}$ du vecteur C_Λ sont exprimées par les coefficients c_j de la forme linéaire du X-problème, les composantes λ_i du vecteur Λ et les éléments $a_{ij}^{(0)}$ de la matrice $A^{(0)}$ des conditions (1.5) selon la formule

$$c_j^{(\Lambda)} = c_j - \sum_{i=1}^{m} \lambda_i a_{ij}^{(0)}, \quad j = 1, 2, \ldots, n. \tag{1.26}$$

A chaque programme de base du Z-problème, on peut faire correspondre un X_Λ-problème qui consiste à maximiser la forme linéaire

$$L_\Lambda(X) = (C_\Lambda, X) = \sum_{j=1}^{n} c_j^{(\Lambda)} x_j \tag{1.27}$$

sous les conditions (1.6)-(1.7). Nous allons voir que la valeur optimale de la forme linéaire (1.27) peut être utilisée pour tester l'optimalité du programme de base du Z-problème qui a engendré le X_Λ-problème correspondant.

En effet, la valeur de la forme linéaire sur les programmes de base du X_Λ-problème (nous les avons désignés par X_ν, $\nu = 1, 2, \ldots, N$) est liée par des relations simples aux critères Δ_ν des vecteurs des conditions du Z-problème:

$$L_\Lambda(X_\nu) = (C_\Lambda, X_\nu) = (C - \Lambda A^{(0)}, X_\nu) = CX_\nu - (\Lambda A^{(0)}), X_\nu).$$

L'utilisation de (1.11) et de (1.12) ainsi que de la formule (1.24) entraîne

$$L_\Lambda(X_\nu) = \sigma_\nu - (\Lambda, P_\nu) = \lambda - \Delta_\nu. \tag{1.28}$$

Une relation analogue a lieu également pour le programme de base optimal du X_Λ-problème:

$$L_\Lambda(X_{\nu*}) = \lambda - \Delta_{\nu*}. \tag{1.29}$$

La condition d'optimalité du programme de base $X_{\nu*}$ s'écrit

$$L_\Lambda(X_{\nu*}) \geqslant L_\Lambda(X_\nu) \quad \text{pour} \quad \nu = 1, 2, \ldots, N,$$

ou, en vertu de (1.28) et (1.29),

$$\Delta_{\nu*} \leqslant \Delta_\nu, \quad \nu = 1, 2, \ldots, N. \tag{1.30}$$

Par conséquent, si $\Delta_{\nu*} \geqslant 0$, alors tout $\Delta_\nu \geqslant 0$. Cela signifie que pour tester l'optimalité du programme de base du Z-problème, il suffit de résoudre le X_Λ -problème qu'il a engendré et de comparer la valeur optimale de $L_\Lambda(X)$ avec le critère λ. (Constatons que l'inégalité $\Delta_{\nu*} > 0$ est impossible, puisque, en substituant à ν dans (1.30) le numéro de l'un quelconque des vecteurs de base du Z-problème, on obtient $\Delta_{\nu*} \leqslant 0$).

Le programme de base étudié du Z-problème est optimal si

$$L_\Lambda(X_{\nu*}) = \lambda.$$

Mais si $L_\Lambda(X_{\nu*}) > \lambda$, il faut passer à un programme de base voisin du Z-problème, à valeur plus grande de la forme linéaire (1.13). L'inégalité (1.30) montre que le vecteur $\overline{P}_k$ associé au programme de base optimal $X_{\nu*}$ admet le plus petit critère relatif

$$\Delta_{\nu*} = \lambda - L_\Lambda(X_{\nu*}) < 0.$$

On introduit ainsi dans la base du Z-problème le vecteur $\overline{P}_k$ tel que

$$\overline{P}_k = \begin{pmatrix} P_k \\ 1 \end{pmatrix} = \begin{pmatrix} A^{(0)}X_{\nu*} \\ 1 \end{pmatrix}. \tag{1.31}$$

Le coefficient correspondant de la forme linéaire (1.13) est égal à

$$\sigma_k = (C, X_{\nu*}). \tag{1.32}$$

Tous les calculs ultérieurs liés à la résolution du Z-problème (test d'irrésolubilité du problème, calcul des éléments de la matrice inverse à la matrice de la base courante, etc.) sont effectués conformément aux règles du deuxième algorithme primal du simplexe.

Prêtons une fois encore attention au fait que pour résoudre le Z-problème (et, par conséquent, pour obtenir le programme optimal du X-problème), nul besoin n'est de connaître d'avance sa matrice des conditions. Il faut connaître seulement les vecteurs des conditions qui entrent dans la base initiale du Z-problème. Les composantes des vecteurs des conditions à introduire dans la base pour améliorer le programme et les coefficients correspondants de la forme linéaire se calculent suivant les formules (1.31) et (1.32), à mesure que la nécessité se présente. Pour résoudre le problème initial, il ne faut généralement calculer qu'une faible partie des programmes de base du $X^{(1)}$-problème et, par suite, une très petite partie des vecteurs des conditions du Z-problème.

Nous avions divisé ci-dessus les conditions du X-problème en deux blocs, aux matrices respectives $A^{(0)}$ et $A^{(1)}$. Le premier bloc contenait m conditions et le deuxième, m_1. Tout ce qui a été dit à propos du X-problème (1.4)-(1.7) peut être repris pour le $X^{(1)}$-problème (1.4), (1.6), (1.7). Le bloc à matrice $A^{(1)}$ peut être divisé en deux avec m et $m_1 - m$ conditions respectivement. La méthode de décomposition peut être appliquée également au $X^{(1)}$-problème (ou plus exactement, à chacun des X_Λ-problèmes). En poursuivant des raisonnements analogues, on aboutit à la conclusion suivante. La résolution des problèmes de programmation linéaire au nombre de conditions quelconque peut être ramenée, en principe, à la discussion de quelques problèmes dont chacun a un nombre de contraintes ne dépassant pas une valeur donnée m.

1.3. Dans la construction du Z-problème, nous étions partis de ce que le domaine défini par les conditions (1.6), (1.7) était borné. Dans le cas contraire, la formule (1.8) perdait son sens.

Une légère modification du Z-problème permet de renoncer à l'hypothèse adoptée.

Dans le cas général, le système de contraintes (1.6), (1.7) découpe dans l'espace des variables X un ensemble convexe G. Le rang du système de contraintes du type (1.6), (1.7) est égal au nombre de composantes du vecteur X. Par suite, d'après le théorème 2.2 du chapitre 5 de [87], tout point de l'ensemble G peut être mis sous la forme

$$X = \sum_{\nu=1}^{N_1} X_\nu z_\nu + \sum_{\nu=N_1+1}^{N} X_\nu z_\nu, \tag{1.33}$$

où

$$\sum_{\nu=1}^{N_1} z_\nu = 1; \tag{1.34}$$

$$z_\nu \geqslant 0, \qquad \nu = 1, 2, \ldots, N. \tag{1.35}$$

Les X_ν sont ici les sommets de l'ensemble convexe G avec $\nu = 1, 2, \ldots, N_1$ et les vecteurs directeurs des arêtes non bornées de cet ensemble avec $\nu = N_1 + 1, \ldots, N$.

Dans le cas considéré, chaque point de l'ensemble G n'est déjà plus une combinaison convexe des vecteurs X_ν.

Pour conserver l'équivalence du X-problème et du Z-problème, il convient de remplacer les conditions (1.15) du Z-problème par une relation du type

$$\sum_{\nu=1}^{N} \varepsilon_\nu z_\nu = 1, \tag{1.36}$$

où

$$\varepsilon_\nu = \begin{cases} 1 \text{ si } X_\nu \text{ correspond à un sommet de l'ensemble connexe} \\ \quad G\ (\nu = 1, 2, \ldots, N_1): \\ 0 \text{ si } X_\nu \text{ correspond à une arête non bornée de l'ensemble} \\ \quad G\ (\nu = N_1 + 1, \ldots, N). \end{cases}$$

Ainsi, dans le cas général le Z-problème se ramène à la maximisation de la forme linéaire

$$\sum_{\nu=1}^{N} \sigma_\nu z_\nu$$

sous les conditions

$$\sum_{\nu=1}^{N} \overline{P}_\nu z_\nu = \overline{B}^{(0)},$$

$$z_\nu \geqslant 0, \quad \nu = 1, 2, \ldots, N,$$

où

$$\overline{P}_\nu = \begin{pmatrix} P_\nu \\ \varepsilon_\nu \end{pmatrix}; \quad P_\nu = A^{(0)} X_\nu; \quad \sigma_\nu = (C, X_\nu). \tag{1.37}$$

Le critère de sélection Δ_ν du vecteur des conditions $\overline{P}_\nu$ par rapport à la base du Z-problème se calcule d'après la relation

$$\Delta_\nu = -L_\Lambda (X_\nu) + \varepsilon_\nu \lambda, \tag{1.38}$$

dont la justification ne diffère en rien de celle de la formule (1.28) donnée plus haut.

L'ensemble convexe G des conditions du $X^{(1)}$-problème, en règle générale, n'étant pas borné, la procédure de résolution du X_Λ-problème successif peut s'achever par le cas 1° ou 2°.

Le cas 1° ou bien correspond à la résolution du Z-problème, ou bien nécessite l'inclusion dans la base successive d'un vecteur $\overline{P}_\nu$ de numéro $\nu \leqslant N_1$ (c'est-à-dire du vecteur correspondant à un sommet de l'ensemble G). Le cas 2° nécessite de poursuivre le processus de résolution du Z-problème ; de plus, il faut introduire dans la base

successive le vecteur $\overline{P}_\nu$ affecté de numéro $\nu > N_1$ (c'est-à-dire le vecteur associé à une arête non bornée de l'ensemble convexe G).

Voici les raisonnements correspondants. Dans le cas 1°, on détermine le programme optimal du X_Λ-problème. Désignons-le par $X_{\nu*}$. Les inégalités

$$L_\Lambda(X_\nu) \leqslant L_\Lambda(X_{\nu*}) \quad \text{pour} \quad \nu = 1, 2, \ldots, N_1$$

et la relation (1.38) conduisent à la condition

$$\Delta_{\nu*} \leqslant \Delta_\nu \quad \text{pour} \quad \nu = 1, 2, \ldots, N_1.$$

Ensuite on vérifie aisément que

$$L_\Lambda(X_\nu) \leqslant 0 \quad \text{pour} \quad \nu = N_1 + 1, \ldots, N. \tag{1.39}$$

En effet, supposons que l'arête non bornée associée à X_ν soit issue du sommet X_{ν_0}. Si $L_\Lambda(X_\nu) > 0$, la valeur de la forme linéaire $L_\Lambda(X_{\nu_0} + \mu X_\nu) = L_\Lambda(X_{\nu_0}) + \mu L_\Lambda(X_\nu)$ sur le programme $X_{\nu_0} + \mu X_\nu$ croît à l'infini lorsque μ augmente. Ainsi, la violation ne serait-ce que de l'une des inégalités (1.39) contredit le fait que le X_Λ-problème est résoluble (on considère le cas 1°).
C'est pourquoi, pour $\nu = N_1 + 1, \ldots, N$,

$$\Delta_\nu = -L_\Lambda(X_\nu) + \varepsilon_\nu \lambda = -L_\Lambda(X_\nu) \geqslant 0.$$

Cela signifie que dans le cas 1°, les critères de sélection des vecteurs des conditions $\overline{P}_\nu$ associés aux arêtes non bornées de l'ensemble convexe G ($\nu = N_1 + 1, \ldots, N$) sont non négatives.

Par suite, dans le cas 1°, il ne faut pas introduire dans la base les vecteurs $\overline{P}_\nu$ ($\nu = N_1 + 1, \ldots, N$).

L'égalité $\Delta_{\nu*} = 0$ signifie que le Z-problème est résolu. Si $\Delta_{\nu*} < 0$, on introduit dans la base suivante le vecteur $\overline{P}_{\nu*}$ correspondant au sommet $X_{\nu*}$ de l'ensemble convexe G.

Le cas 2°, qui peut se présenter lors de la résolution du X_Λ-problème, signifie qu'à une certaine itération le critère $\Delta_j^{(\Lambda)}$ d'un vecteur des conditions du X_Λ-problème se trouve être négatif et que, de plus, toutes les composantes x_{ij} de la décomposition de ce vecteur par rapport aux vecteurs de la base sont non positives. Supposons que les vecteurs de la base du programme de base précédent X_{ν_0} soient les p premiers vecteurs du X_Λ-problème (p est le nombre de conditions indépendantes du X_Λ-problème). Il vient alors

$$X_{\nu_0} = (x_{10}, x_{20}, \ldots, x_{p0}, 0, 0, \ldots, 0).$$

Considérons le vecteur

$$X_\nu = (\underbrace{-x_{1j}, \ldots, -x_{pj}, 0, \ldots, 1}_{j}, \ldots, 0).$$

Par construction de X_ν,

$$A^{(1)} X_\nu = 0.$$

Par ailleurs, en raison des conditions rendant la solution du X_Λ-problème ($x_{ij} \leqslant 0$) irréalisable

$$X_\nu \geqslant 0.$$

Par conséquent, le vecteur

$$X = X_{\nu_0} + \mu X_\nu$$

est un programme du X_Λ-problème quel que soit $\mu \geqslant 0$. Ceci signifie que le vecteur X_ν est un vecteur directeur d'une arête non bornée de l'ensemble convexe (1.6), (1.7). La valeur de la forme linéaire du X_Λ-problème sur le vecteur X_ν est égale à

$$L_\Lambda(X_\nu) = -\sum_{i=1}^{p} c_i^{(\Lambda)} x_{ij} + c_j^{(\Lambda)}.$$

D'autre part, la définition du critère de sélection du vecteur des conditions du X_Λ-problème amène

$$\Delta_j^{(\Lambda)} = \sum_{i=1}^{p} c_i^{(\Lambda)} x_{ij} - c_j^{(\Lambda)}.$$

Par conséquent,

$$L_\Lambda(X_\nu) = -\Delta_j^{(\Lambda)}. \tag{1.40}$$

Mais en raison de (1.38) on a pour le vecteur directeur X_ν d'une arête non bornée de l'ensemble convexe ($\varepsilon_\nu = 0$)

$$\Delta_\nu = -L_\Lambda(X_\nu).$$

En comparant cette relation avec l'égalité (1.40), on a

$$\Delta_\nu = \Delta_j^{(\Lambda)}. \tag{1.41}$$

Dans le cas 2°, le critère $\Delta_j^{(\Lambda)}$ est négatif. C'est pourquoi, d'après la formule (1.41), on peut introduire dans la base successive du Z-problème le vecteur

$$\bar{P}_\nu = \begin{pmatrix} A^{(0)} X_\nu \\ 0 \end{pmatrix},$$

associé à une arête non bornée du domaine de définition du X_Λ-problème.

Ainsi, lorsque la résolution du X_Λ-problème conduit au cas 2°, la procédure de résolution du Z-problème doit être poursuivie. Le vecteur à ajouter à la base est défini par celui des vecteurs des conditions du X_Λ-problème qui vérifie les restrictions du cas 2°.

1.4. Voici quelques observations sur la détermination du programme de base initial du Z-problème.

Le programme de base initial du Z-problème se calcule sans peine dans les cas où les contraintes du X-problème ont toutes la forme d'inégalités aux deuxièmes membres non négatifs. Dans ce cas, les conditions du Z-problème s'écrivent

$$\sum_{\nu=1}^{N} P_\nu z_\nu \leqslant B^{(0)}, \tag{1.42}$$

$$\sum_{\nu=1}^{N} \varepsilon_\nu z_\nu = 1,$$

$$z_\nu \geqslant 0, \quad \nu = 1, 2, \ldots, N.$$

L'ensemble convexe des conditions du $X^{(1)}$-problème correspondant a le sommet

$$X_0 = (\underbrace{0, \ldots, 0}_{n}).$$

A ce sommet correspond le vecteur des conditions du Z-problème

$$\bar{P}_0 = \begin{pmatrix} P_0 \\ 1 \end{pmatrix} = \begin{pmatrix} A^{(0)} X_0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0_m \\ 1 \end{pmatrix}.$$

En introduisant dans le système d'inégalités (1.42) m variables supplémentaires $z'_i \geqslant 0$ ($i = 1, 2, \ldots, m$), on obtient la collection suivante de vecteurs des conditions, qui peut servir de base initiale du Z-problème :

$$(\bar{P}_0, \bar{P}'_1, \bar{P}'_2, \ldots, \bar{P}'_m) = \left\| \begin{matrix} 0 & 1 & 0 \ldots 0 \\ 0 & 0 & 1 \ldots 0 \\ . & . & . \; . \; . \; . \; . \\ 0 & 0 & 0 \ldots 1 \\ 1 & \underbrace{0 \quad 0 \ldots 0}_{m} \end{matrix} \right\| = \left\| \begin{matrix} 0 & E_m \\ 1 & 0 \end{matrix} \right\|.$$

E_m est ici la matrice unité d'ordre m.

Les variables de base correspondant au programme de base initial du Z-problème sont égales respectivement à

$$1, \; b_1^{(0)}, \; b_2^{(0)}, \; \ldots, \; b_m^{(0)},$$

où les $b_i^{(0)}$ sont les composantes du vecteur $B^{(0)}$.

La matrice inverse relative à la base initiale est égale à

$$\left\| \begin{matrix} 0 & 1 \\ E_m & 1 \end{matrix} \right\|.$$

Si le X-problème est mis sous forme canonique, la définition du programme de base initial donne lieu à un volume de calculs

bien plus grand. Il faut alors faire appel à l'un des procédés décrits au § 6 du chapitre 7 de [87], où l'on trouve les méthodes de calcul du programme de base initial.

Il est parfois plus simple de déterminer le programme du $X^{(1)}$-problème que son programme de base. Il en est ainsi, par exemple, dans les cas où, pour quelque raison que ce soit, il est rationnel de résoudre le $X^{(1)}$-problème par la méthode itérative. Dans des cas semblables, il est utile de tenir compte des considérations suivantes.

Soit ξ_μ ($\mu = 1, 2, \ldots, q$) le programme du $X^{(1)}$-problème (qui n'est pas, en général, un programme de base). Introduisons les notations

$$\sigma'_\mu = (C, \xi_\mu), \quad \bar{P}'_\mu = \begin{pmatrix} A^{(0)}\xi_\mu \\ 1 \end{pmatrix}.$$

On voit aisément que le problème de maximisation de la forme linéaire

$$\sum_{\nu=1}^{N} \sigma_\nu z_\nu + \sum_{\mu=1}^{q} \sigma'_\mu z'_\mu \tag{1.43}$$

sous les conditions:

$$\sum_{\nu=1}^{N} \bar{P}_\nu z_\nu + \sum_{\mu=1}^{q} \bar{P}'_\mu z'_\mu = \bar{B}^{(0)}, \tag{1.44}$$

$$\left.\begin{aligned} z_\nu &\geqslant 0, \quad \nu = 1, 2, \ldots, N, \\ z'_\mu &\geqslant 0, \quad \mu = 1, 2, \ldots, q, \end{aligned}\right\} \tag{1.45}$$

est équivalent au Z-problème. C'est pourquoi on peut prendre comme programme de base initial du Z-problème n'importe quel système linéairement indépendant de $m + 1$ vecteurs des conditions du problème (1.43)-(1.45) jouissant de cette propriété que les composantes de décomposition du vecteur des contraintes $\bar{B}^{(0)}$ par rapport aux vecteurs de ce système sont non négatives.

1.5. Ebauchons maintenant le schéma de calcul de la méthode de décomposition.

La résolution du problème débute par la division du système des conditions du problème en blocs dont les dimensions sont déterminées par la capacité de la mémoire à accès immédiat de la machine utilisée pour le calcul du programme optimal. Lors de la division de la matrice A des conditions du problème en matrices $A^{(0)}$ et $A^{(1)}$, il est logique d'inclure dans le deuxième bloc les conditions conduisant à la matrice $A^{(1)}$ la plus « rare » possible. Ainsi on réussit généralement à simplifier le calcul des programmes optimaux des X_Λ-problèmes aux différentes itérations de la résolution du Z-problème.

Comme on l'a déjà indiqué, le Z-problème se résout d'après les règles du deuxième algorithme primal du simplexe. Toutefois, les

particularités de la méthode de décomposition contraignent d'apporter certaines modifications au schéma de calcul.

Lors de la résolution du Z-problème, on ne remplit que les tableaux principaux I. La nécessité d'un tableau auxiliaire disparaît. A la place de ce dernier, il est avantageux d'introduire le tableau II où figureront les lignes de la matrice des conditions $A^{(0)}$, les coefficients c_j de la forme linéaire du X-problème initial, les coefficients $c_j^{(\Lambda)}$ des formes linéaires et les programmes optimaux (ainsi que les vecteurs directeurs des arêtes non bornées du domaine de définition) des X_Λ-problèmes, nécessaires pour la réalisation des itérations successives de la résolution du Z-problème. Dans la dernière ligne du tableau II sont placées les composantes de la solution X^* du problème initial.

Les coefficients $c_j^{(\Lambda)}$ se calculent suivant les formules (1.26) où les $a_{ij}^{(0)}$ sont les éléments du bloc $A^{(0)}$ et les λ_i, les éléments de la $(m+2)$-ième ligne du tableau principal du Z-problème.

Supposons que la l-ième itération du processus de résolution du Z-problème soit terminée. Il en résulte le l-ième tableau principal du Z-problème (sauf les deux dernières colonnes $\overline{P}_k$ et θ). En se servant des quantités λ_i, obtenues à la l-ième itération, on compose la forme linéaire du X_Λ-problème. Il est bon d'effectuer aussi la résolution des X_Λ-problèmes selon les règles de la méthode primale du simplexe. On peut utiliser de plus le programme optimal du X_Λ-problème associé à la l-ième itération comme programme de base initial du X_Λ-problème obtenu à la $(l+1)$-ième itération. Un tel ordre de calcul diminue généralement le nombre d'itérations nécessaires pour résoudre la suite des X_Λ-problèmes. Le choix du premier ou du deuxième algorithme de la méthode primale du simplexe est défini par les particularités de structure de la matrice $A^{(1)}$ et se fait en se guidant par les recommandations de [87] (§ 5 du chapitre 8).

Dans le schéma de calcul de la méthode de décomposition il est commode de prévoir un tableau spécial de récapitulation (tableau III) dans lequel sont notés les principaux paramètres caractérisant chaque itération de la procédure de résolution du Z-problème. Dans ce tableau sont portées les valeurs de λ [$(m+1)$-ième composante du vecteur $\overline{\Lambda}$], les valeurs optimales L_Λ^* des formes linéaires des X_Λ-problèmes, les oritères $\Delta^* = \Delta_k$ des vecteurs des conditions $\overline{P}_k$ ajoutés à la base du Z-problème, les coefficients correspondants σ_k de la forme linéaire du Z-problème et les composantes des vecteurs $\overline{P}_k$. Chaque itération de la résolution du Z-problème occupe une ligne distincte du tableau récapitulatif.

On inscrit dans la colonne $\overline{P}_k$ des tableaux principaux du Z-problème les composantes de la décomposition du vecteur $\overline{P}_k$ par rapport aux vecteurs de la base courante. La colonne θ des tableaux prin-

cipaux se compose d'après les règles générales du deuxième algorithme de la méthode primale du simplexe. Ensuite, selon l'usage, on détermine le vecteur à éliminer de la base du Z-problème et l'on remplit selon les formules de récurrence connues la partie principale du tableau principal associé à la nouvelle itération. Après un nombre fini d'étapes, on aboutit à la solution du Z-problème ou bien l'on établit que sa solution est irréalisable (forme linéaire infinie sur l'ensemble des programmes possibles). La forme linéaire du Z-problème ne peut être infinie dans le domaine de sa définition que dans le cas où le X-problème est irréalisable.

Le programme optimal du X-problème est déterminé par la formule (1.19) à l'aide des composantes z^*_ν de la solution du Z-problème et des programmes optimaux X_ν des X_Λ-problèmes successifs.

Le schéma de calcul suppose naturellement l'existence des tableaux qui fixent la procédure de résolution de la suite des X_Λ -problèmes.

1.6. Illustrons l'ordre des calculs sur l'exemple suivant.

Soit la forme linéaire à maximiser

$$L(X) = 2x_1 + 6x_2 + 5x_3 - 3x_5 + 8x_6 \tag{1.46}$$

sous les conditions

$$\left.\begin{aligned} -4x_1 + 8x_2 + 6x_3 - x_4 + 5x_5 + x_6 &\leqslant 8, \\ 2x_1 + 3x_2 + 5x_3 + 4x_4 + 6x_5 + 6x_6 &\leqslant 15, \end{aligned}\right\} \tag{1.47}$$

$$\left.\begin{aligned} -3x_1 + 3x_2 + 2x_4 + 3x_6 &\leqslant 4, \\ 2x_1 - 9x_2 + 9x_3 + 3x_4 + x_5 + x_6 &\leqslant 3, \\ 6x_2 + 8x_3 - 3x_4 + 3x_5 - 2x_6 &\leqslant 6, \end{aligned}\right\} \tag{1.48}$$

$$x_j \geqslant 0,\ j = 1, 2, \ldots, 6. \tag{1.49}$$

D'après une contrainte supplémentaire, la solution du problème doit être ramenée à la recherche de la solution des problèmes de programmation linéaire dont le nombre de conditions ne dépasse pas trois.

Les conditions du problème peuvent être divisées en blocs $A^{(0)}$ et $A^{(1)}$ de façon arbitraire, puisqu'il n'y a pas lieu d'escompter que l'on puisse utiliser les traits propres à l'une quelconque des contraintes pour simplifier la procédure de résolution. Pour fixer les idées, rapportons les deux premières contraintes au bloc $A^{(0)}$ et les trois suivantes au bloc $A^{(1)}$.

Le Z-problème correspondant compte trois conditions dont deux sont engendrées par les contraintes (1.47) et les programmes de base du système (1.48), (1.49). La troisième condition est de la forme (1.15).

Selon les observations du paragraphe 1.4, la base initiale du Z-problème se compose de trois vecteurs linéairement indépendants $\overline{P}_0$, $\overline{P}'_1$, $\overline{P}'_2$. Le vecteur des conditions $\overline{P}_0$ est associé au programme de base X_0 du système (1.48), (1.49),

$$X_0 = (0, 0, 0, 0, 0, 0).$$

D'après les formules (1.11), (1.12) et (1.18), il vient

$$\sigma_0 = 0, \quad P_0 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad \overline{P}_0 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}.$$

Les vecteurs $\overline{P}_1'$ et $\overline{P}_2'$ sont les vecteurs des conditions du Z-problème correspondant aux variables supplémentaires (variables d'écart) qui transforment en égalités les deux premières conditions-inégalités de ce problème.

Par conséquent,

$$\overline{P}_1' = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \quad \overline{P}_2' = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}.$$

Les coefficients de la forme linéaire du Z-problème associés aux vecteurs des conditions $\overline{P}_1'$ et $\overline{P}_2'$ sont nuls.

Les vecteurs $\overline{P}_0$, $\overline{P}_1'$ et $\overline{P}_2'$ de la base correspondent aux variables de base

$$z_1 = z_{10} = 1, \quad z_2 = z_{20} = 8, \quad z_3 = z_{30} = 15.$$

La matrice inverse de la matrice de base est de la forme

$$\begin{Vmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{Vmatrix}.$$

Les critères λ_i de toutes les conditions du Z-problème par rapport à la base initiale sont égaux à zéro, puisque tout σ_ν qui correspond aux variables de base a une valeur nulle. Ainsi,

$$\lambda_1 = \lambda_2 = \lambda = 0.$$

C'est pourquoi la forme linéaire du X_Λ-problème associée à la base initiale du Z-problème coïncide avec la forme linéaire (1.46) du X-problème.

Cherchons la solution du X_Λ-problème suivant les règles du deuxième algorithme de la méthode primale du simplexe, le vecteur X_0 étant pris comme programme de base initial. Au bout de trois itérations, on obtient la solution du X_Λ-problème. Désignons ce programme de base par X_1:

$$X_1 = (25;\ 7{,}333;\ 0;\ 0;\ 0;\ 19).$$

La valeur de la forme linéaire sur X_1 est

$$L_\Lambda^* = L_\Lambda(X_1) = 246.$$

D'après la formule (1.29),

$$\Delta_{\nu^*} = \Delta_1 = 0 - 246 = -246.$$

La relation $\Delta_1 < 0$ atteste que le programme de base du Z-problème peut être amélioré. A cette fin il faut ajouter à la base du Z-problème le vecteur $\overline{P}_1$ associé au programme de base X_1. Suivant les formules (1.11), (1.12) et (1.18), on obtient

$$\sigma_1 = 246; \quad P_1 = \begin{pmatrix} -22{,}333 \\ 186 \end{pmatrix}; \quad \overline{P}_1 = \begin{pmatrix} -22{,}333 \\ 186 \\ 1 \end{pmatrix}.$$

En procédant ensuite conformément aux règles du deuxième algorithme de la méthode primale du simplexe on retranche de la base le vecteur $\overline{P}_2'$. L'itération suivante du Z-problème donne les critères de ses conditions par rapport à la base $\overline{P}_0$, $\overline{P}_1'$, $\overline{P}_1$:

$$\lambda_1 = 0; \quad \lambda_2 = 1{,}323, \quad \lambda = 0.$$

Tableau 4.1

I

Z-problème

	N	σ_k	B_z	$e_0 = z_{i0}$	e_1	e_2	e_3	$\overline{P}_k$	θ	N_z
	1		$\overline{P}_0$	1			1	1	1	
	2		$\overline{P}'_1$	8	1			−22,333	—	
←	3		$\overline{P}'_2$	15		1		186	$\boxed{0,0806}$	0
	4	—	—					−246	—	
←	1		$\overline{P}_0$	0,919		−0,00538	1	0,974	$\boxed{0,943}$	
	2		$\overline{P}'_1$	9,801	1	0,120		9,487	1,033	1
→	3	246	$\overline{P}_1$	0,0806		0,00538		0,0255	3,158	
	4	—	—	19,839		1,323		−2,218	—	
→	1	8,5	$\overline{P}'_2$	0,943		−0,00552	1,026			
	2		$\overline{P}'_1$	0,851	1	0,172	−9,736			2
	3	246	$\overline{P}_1$	0,0566		0,00552	−0,0262			
	4	—	$L_z^* = 21,931$			1,310	2,276		—	

Tableau 4.2

II

N	$B^{(0)}$	A_1	A_2	A_3	A_4	A_5	A_6
1	8	−4	8	6	−1	5	1
2	15	2	3	5	4	6	6
3	c_j	2	6	5		−3	8
4	c_j^{Λ}	−0,645	2,032	−1,613	−5,290	−10,935	0,0645
5	c_j^{Λ}	−0,621	2,069	−1,552	−5,241	−10,862	0,138
6	X_1	25	7,333				19
7	X_2		1,083				0,25
	X^*	1,415	1,437				1,310

Tableau 4.3

III

N_z	λ	L_{Λ}^{*}	$\Delta_{\nu^*} = \Delta_k$	σ_k	$\overline{P}_k$	Composantes de $\overline{P}_k$		
0	0	246	−246	246	$\overline{P}_1$	−22,333	186	1
1	0	2,218	−2,218	8,5	$\overline{P}_2$	8,914	4,75	1
2	2,276	2,276	0					

D'après les formules (1.26), la forme linéaire du X_Λ-problème successif s'écrit

$$-0{,}645x_1 + 2{,}032x_2 - 1{,}613x_3 - 5{,}290x_4 - 10{,}935x_5 + 0{,}0645x_6. \quad (1.50)$$

Prenons pour programme de base initial du X_Λ-problème la solution du X_Λ-problème précédent. Le programme optimal du problème s'obtient en une itération. Désignons-le par X_2:

$$X_2 = (0;\ 1{,}083;\ 0;\ 0;\ 0;\ 0{,}25).$$

La valeur correspondante de la forme linéaire du X_Λ-problème est

$$L_\Lambda^* = L_\Lambda(X_2) = 2{,}218.$$

Le critère

$$\Delta_{\nu*} = \Delta_2 = \lambda - L_\Lambda^*(X_2) = -2{,}218$$

est négatif. Par conséquent le programme du Z-problème peut être amélioré. On introduit dans la base le vecteur

$$\overline{P}_2 = \begin{pmatrix} A^{(0)}X_2 \\ 1 \end{pmatrix} = \begin{pmatrix} 8{,}917 \\ 4{,}75 \\ 1 \end{pmatrix}.$$

L'itération suivante du Z-problème conduit à un X_Λ-problème de forme linéaire

$$-0{,}621x_1 + 2{,}069x_2 - 1{,}552x_3 - 5{,}241x_4 - 10{,}862x_5 + 0{,}138x_6. \quad (1.51)$$

On voit que les coefficients de la forme linéaire (1.51) diffèrent peu des coefficients correspondants de la forme linéaire (1.50) du X_Λ-problème précédent. La vérification montre immédiatement que la solution du dernier X_Λ-problème coïncide avec le programme optimal du problème précédent, soit

$$X_3 = X_2.$$

Dans ces conditions, la valeur de la forme linéaire (1.51) sur le programme X_3 est

$$L_\Lambda^* = L_\Lambda(X_3) = 2{,}276$$

et

$$\Delta_{\nu*} = \Delta_3 = 0.$$

D'après le critère d'optimalité, le programme de base du Z-problème qui a engendré le dernier X_Λ-problème est solution du Z-problème.

La base optimale du Z-problème se compose de deux vecteurs des conditions $\overline{P}_1$ et $\overline{P}_2$ et du vecteur $\overline{P}_1'$ associé à l'une des variables supplémentaires qui en s'ajoutant aux conditions-inégalités les transforment en égalités.

Les vecteurs $\overline{P}_1$ et $\overline{P}_2$ de la base sont associés aux composantes de base

$$z_1^* = z_{30} = 0{,}0566, \quad z_2^* = z_{10} = 0{,}943.$$

D'après la formule (1.19), le programme optimal du X-problème s'écrit

$$X^* = z_1^* X_1 + z_2^* X_2.$$

Les calculs donnent la solution du problème initial

$$X^* = (1{,}415;\ 1{,}437;\ 0;\ 0;\ 0;\ 1{,}310).$$

Toute la procédure de résolution du problème est portée sur les tableaux 4.1 à 4.5. Le tableau 4.1 est composé de tableaux principaux I de la résolution du Z-problème. Le tableau 4.2 contient les paramètres du bloc $A^{(0)}$ (constantes des conditions (1.47) du X-problème), les coefficients c_j de sa forme linéaire, les

coefficients $c_j^{(\Lambda)}$ des formes linéaires, les composantes des programmes optimaux des X_Λ-problèmes successifs, les composantes de la solution du X-problème.

Dans le tableau 4.3 figurent les paramètres principaux de chaque itération du Z-problème (valeurs λ des critères de sélection de la dernière condition du Z-problème par rapport à ses bases, valeurs L_Λ^* des formes linéaires des X_Λ-problèmes sur leurs programmes optimaux, critères Δ_{ν^*}, composantes $\overline{P}_{i\,(k)}$ des vecteurs des conditions $\overline{P}_k$ ajoutés à la base et coefficients σ_k de la forme linéaire du Z-problème qui leur correspondent).

Les tableaux 4.4 et 4.5 représentent les tableaux principaux et auxiliaires qui traduisent la procédure de résolution de la suite des X_Λ-problèmes.

§ 2. Cas particuliers et modifications de la méthode de décomposition

2.1. La méthode de décomposition, décrite au paragraphe précédent et appliquée au problème général de programmation linéaire, est particulièrement efficace lorsque la matrice $A^{(1)}$ a une structure diagonale par blocs. Le cas considéré par Dantzig et Wolfe était précisément celui de

$$A^{(1)} = \begin{Vmatrix} \hat{A}^{(1)} & 0 & \dots & 0 \\ 0 & \hat{A}^{(2)} & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & \hat{A}^{(s)} \end{Vmatrix},$$

où $\hat{A}^{(t)} = \| \hat{a}_{ij}^{(t)} \|$ est une matrice de dimension $m_t \times n_t$ *).

La structure diagonale par blocs de la matrice $A^{(1)}$ simplifie la résolution des X_Λ-problèmes et justifie la division des conditions du X-problème en deux groupes.

Dans le cas de la matrice diagonale par blocs $A^{(1)}$, il est commode de mettre le X-problème sous la forme

$$L = \sum_{t=1}^{s} (C^{(t)}, X^{(t)}) \to \max \tag{2.1}$$

sous les conditions

$$\sum_{t=1}^{s} A_t^{(0)} X^{(t)} = B^{(0)}, \tag{2.2}$$

$$A^{(t)} X^{(t)} = B^{(t)}, \quad t = 1, 2, \dots, s, \tag{2.3}$$

$$X^{(t)} \geqslant 0, \quad t = 1, 2, \dots, s. \tag{2.4}$$

$X^{(t)}$ est ici un vecteur-colonne de dimension n_t; $A_t^{(0)} = \| A_{ij}^{(0)(t)} \|$, une matrice de dimension $m \times n_t$; $A^{(t)} = \| a_{ij}^{(t)} \|$, une matrice de dimension $m_t \times n_t$; $B^{(0)}$, un vecteur-colonne de dimension m;

*) Pour simplifier l'écriture, remplaçons la notation $\hat{A}^{(t)}$ des sous-matrices par $A^{(t)}$. Ceci ne prêtera pas à confusion, puisqu'il ressort toujours du texte de quelle matrice $A^{(1)}$ il s'agit. Il convient de faire une observation analogue à l'égard de la notation du $X^{(1)}$-problème donnée plus bas.

Tableau 4.4

X_Λ-problèmes

$\boxed{0}$	N	$c_j^{(\Lambda)}$	B	e_0	e_1	e_2	e_3	A_k	θ	N_Λ
←	1		A_7	4	1			3	$\boxed{1,333}$	
	2		A_8	3		1		1	3	0
	3		A_9	6			1	−2	—	
	4	—	—					−8	—	
→	1	8	A_6	1,333	0,333			−1	—	
←	2		A_8	1,667	−0,333	1		3	$\boxed{0,556}$	1
	3		A_9	8,667	0,667		1	−2	—	
	4	—	—	10,667	2,667			−10	—	
	1	8	A_6	1,889	0,222	0,333		−2,333	—	
→	2	2	A_1	0,556	−0,111	0,333		−3,333	—	2
←	3		A_9	9,778	0,444	0,667	1	1,333	$\boxed{7,333}$	
	4	—	—	16,222	1,556	3,333		−31,333	—	
	1	8	A_6	19	1	1,5	1,75			
	2	2	A_1	25	1	2	2,5			3
→	3	6	A_2	7,333	0,333	0,5	0,75			
	4	—	$L_\Lambda^* = 246$		12	19	23,5		—	

$\boxed{1}$	N	$c_j^{(\Lambda)}$	B	e_0	e_1	e_2	e_3	A_k	θ	N_Λ
	1	0,0645	A_6	19	1	1,5	1,75	1,5	12,667	
←	2	−0,0645	A_1	25	1	2	2,5	2	$\boxed{12,5}$	0
	3	2,032	A_2	7,333	0,333	0,5	0,75	0,5	14,667	
	4	—	—	0	0,0968	−0,177	0,0242	−0,177	—	
	1	0,0645	A_6	0,25	0,25		−0,125			
→	2		A_8	12,5	0,5	1	1,25			1
	3	2,032	A_2	1,083	0,0833		0,125			
	4	—		$L_\Lambda^* = 2,218$	0,185		0,246		—	

$\boxed{2}$	N	$c_j^{(\Lambda)}$	B	e_0	e_1	e_2	e_3	A_k	θ	N_Λ
	1	0,138	A_6	0,25	0,25		−0,125			
	2		A_8	12,5	0,5	1	1,25			
	3	2,069	A_2	1,083	0,0833		0,125			0
	4	—		$L_\Lambda^* = 2,276$	0,207		0,241		—	

Tableau 4.5

N	$\bar{B}^0$	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	
1	4	−3	3		2		3	1			
2	3	2	−9	9	3	1	1		1		
3	6		6	8	−3	3	−2			1	
4		2	6	5		−3	8				N_z
0	Δ	−2	−6	−5		3	$\boxed{-8}$				
1	Δ′	$\boxed{-10}$	2	−5	5,333	3		2,667			0
2	Δ″		$\boxed{-31,333}$	25	13,111	6,333		1,556	3,333		
3	Δ‴			354	10,5	92,5		12	19	23,5	
—	c_j	−0,645	2,032	−1,613	−5,290	−10,935	0,0645				
0	Δ			0,210	4,879	10,831		0,0968	$\boxed{-0,177}$	0,0242	1
1	Δ′	0,0887		3,581	4,923	11,673		0,185		0,246	
0	c_j	−0,621	2,069	−1,552	−5,241	−10,862	0,138				2
0	Δ			3,483	4,931	11,586		0,207		0,241	

$B^{(t)}$, un vecteur-colonne de dimension m_t; $C^{(t)}$, un vecteur-ligne de dimension n_t.

Dans le cas considéré tout point X de l'ensemble convexe G des conditions du $X^{(1)}$-problème peut être présenté sous la forme

$$X=(X^{(1)}, X^{(2)}, \ldots, X^{(s)}),\ X^{(t)}=\sum_{\nu=1}^{N_t} z_\nu^{(t)} X_\nu^{(t)}, \quad t=1, 2, \ldots, s. \tag{2.5}$$

Les $X_\nu^{(t)}$ correspondent ici aux sommets et aux vecteurs directeurs des arêtes non bornées des ensembles convexes G_t des conditions des $X^{(t)}$-problèmes obtenus par partition du $X^{(1)}$-problème:

$$z_\nu^{(t)} \geqslant 0, \quad \nu = 1, 2, \ldots, N_t; \quad t = 1, 2, \ldots, s;$$

$$\sum_{\nu=1}^{N_t} \varepsilon_\nu^{(t)} z_\nu^{(t)} = 1, \quad t=1, 2, \ldots, s;$$

$$\varepsilon_\nu^{(t)} = \begin{cases} 1 \text{ si } X_\nu^{(t)} \text{ est associé à un sommet de l'ensemble convexe} \\ \quad G_t \text{ des conditions du } X^{(t)}\text{-problème } (\nu = 1, 2, \ldots \\ \quad \ldots, N_t^{(1)};\ t = 1, 2, \ldots, s); \\ 0 \text{ si } X_\nu^{(t)} \text{ est associé à une arête non bornée de l'ensemble} \\ \quad G_t\ (\nu = N_t^{(1)} + 1, \ldots, N_t;\ t = 1, 2, \ldots, s). \end{cases}$$

Les $X^{(t)}$-problèmes ($t = 1, 2, \ldots, s$) obtenus par partition du problème $X^{(1)}$ sont de la forme

$$L^{(t)} = (C^{(t)}, X^{(t)}) \to \max \tag{2.6}$$

sous les conditions

$$A^{(t)} X^{(t)} = B^{(t)}, \tag{2.7}$$

$$X^{(t)} \geqslant 0. \tag{2.8}$$

Le problème initial engendre un Z-problème, qui peut dans notre cas être mis sous la forme:

$$\sum_{t=1}^{s} \sum_{\nu=1}^{N_t} \sigma_\nu^{(t)} z_\nu^{(t)} \to \max, \tag{2.9}$$

$$\sum_{t=1}^{s} \sum_{\nu=1}^{N_t} P_\nu^{(t)} z_\nu^{(t)} = B^{(0)}, \tag{2.10}$$

$$\sum_{\nu=1}^{N_t} \varepsilon_\nu^{(t)} z_\nu^{(t)} = 1, \quad t=1, 2, \ldots, s, \tag{2.11}$$

$$z_\nu^{(t)} \geqslant 0, \quad \nu = 1, 2, \ldots, N_t; \quad t = 1, 2, \ldots, s, \tag{2.12}$$

où

$$\sigma_\nu^{(t)} = (C^{(t)}, X_\nu^{(t)}); \tag{2.13}$$

$$P_\nu^{(t)} = A_t^{(0)} X_\nu^{(t)}. \tag{2.14}$$

Il est parfois commode de remplacer les conditions (2.10) et (2.11) du Z-problème par un système de contraintes

$$\sum_{t=1}^{s} \sum_{\nu=1}^{N_t} \overline{P}_\nu^{(t)} z_\nu^{(t)} = \overline{B}^{(0)}, \tag{2.15}$$

où

$$\overline{P}_\nu^{(t)} = \begin{pmatrix} P_\nu^{(t)} \\ \varepsilon_\nu^{(t)} e_t \end{pmatrix}; \tag{2.16}$$

$$\overline{B}^{(0)} = \begin{pmatrix} B^{(0)} \\ I_s \end{pmatrix}; \tag{2.17}$$

e_t est le vecteur unité de dimension s avec l'unité figurant à la t-ième place;

I_s, le vecteur de dimension s à composantes unités.

La solution du Z-problème détermine univoquement le programme optimal du X-problème. Si les nombres $z_\nu^{*(t)}$ définissent la solution du Z-problème, les vecteurs

$$\overset{*}{X}{}^{(t)} = \sum_{\nu=1}^{N_t} \overset{*}{z}{}_\nu^{(t)} X_\nu^{(t)} \tag{2.18}$$

constituent le programme optimal

$$X^* = (\overset{*}{X}{}^{(1)}, \overset{*}{X}{}^{(2)}, \ldots, \overset{*}{X}{}^{(s)})$$

du problème initial.

Les X_Λ-problèmes liés à chaque étape de la procédure de résolution du Z-problème se décomposent en s $X_\Lambda^{(t)}$-problèmes (en fonction du nombre de blocs). La solution d'un X_Λ-problème se compose des solutions des $X_\Lambda^{(t)}$-problèmes. De plus,

$$L_\Lambda(X^*) = \sum_{t=1}^{s} L_\Lambda^{(t)}(\overset{*}{X}{}^{(t)}).$$

Les valeurs optimales des formes linéaires des $X_\Lambda^{(t)}$-problèmes sont utilisées pour tester l'optimalité du programme de base du Z-problème et pour reformer la base du Z-problème si le programme exploré n'est pas optimal.

Considérons plus près l'ordre de formation d'un $X_\Lambda^{(t)}$-problème et de construction du critère d'optimalité du programme de base du Z-problème.

A chaque itération du Z-problème, on détermine le vecteur $\overline{\Lambda}$ (de dimension $(m + s)$) des critères de sélection des conditions du problème par rapport à la base du programme de base étudié.

Divisons les composantes du vecteur $\overline{\Lambda}$ en deux groupes, conformément aux deux groupes de composantes des vecteurs des conditions $\overline{P}_\nu^{(t)}$ [cf. (2.16)].

Le vecteur $\overline{\Lambda}$ des critères de sélection des conditions est associé aux vecteurs de la base du programme de base correspondant du Z-problème par des égalités du type

$$(\overline{\Lambda},\ \overline{P}_\nu^{(t)}) = \sigma_\nu^{(t)}$$

ou

$$(\Lambda^{(1)},\ P_\nu^{(t)}) + (\Lambda^{(2)},\ \varepsilon_\nu^{(t)} e_t) = \sigma_\nu^{(t)}, \tag{2.19}$$

où le vecteur $\overline{\Lambda}^{(1)} = (\lambda_1^{(1)},\ \lambda_2^{(1)},\ \ldots,\ \lambda_m^{(1)})$ se compose des m premières composantes du vecteur $\overline{\Lambda}$ [des critères des conditions (2.10)], et où le vecteur $\overline{\Lambda}^{(2)} = (\lambda_1^{(2)},\ \ldots,\ \lambda_s^{2})$ est constitué des s composantes qui suivent [des critères des conditions (2.11)].

Ainsi, aux vecteurs $\overline{P}_\nu^{(t)}$ de la base courante du Z-problème correspondent les relations

$$(\overline{\Lambda}^{(1)},\ P_\nu^{(t)}) + \varepsilon_\nu^{(t)} \lambda_t^{(2)} = \sigma_\nu^{(t)}. \tag{2.20}$$

Le vecteur $\overline{\Lambda}^{(1)}$ est utilisé pour construire les $X_\Lambda^{(t)}$-problèmes. Chaque $X_\Lambda^{(t)}$-problème ne se distingue de $X^{(t)}$-problème (2.6)-(2.8) correspondant que par la forme linéaire. Le $X_\Lambda^{(t)}$-problème consiste à maximiser la forme linéaire

$$L_\Lambda^{(t)}(X^{(t)}) = (C_\Lambda^{(t)},\ X^{(t)}) = (C^{(t)} - \Lambda^{(1)} A_t^{(0)},\ X^{(t)}) \tag{2.21}$$

sous les conditions (2.7), (2.8).

Les critères $\Delta_\nu^{(t)}$ des vecteurs des conditions $\overline{P}_\nu^{(t)}$ par rapport à la base courante du Z-problème sont associés par des relations simples aux valeurs de la forme linéaire (2.21) sur les programmes de base des $X_\Lambda^{(t)}$-problèmes et les vecteurs directeurs des arêtes infinies de l'ensemble G_t. Cette circonstance peut être mise à profit pour construire le critère d'optimalité du programme de base du Z-problème. On a

$$\Delta_\nu^{(t)} = (\overline{\Lambda},\ \overline{P}_\nu^{(t)}) - \sigma_\nu^{(t)}$$

ou, ce qui revient au même,

$$\Delta_\nu^{(t)} = (\overline{\Lambda}^{(1)},\ P_\nu^{(t)}) + \varepsilon_\nu^{(t)} \lambda_t^{(2)} - \sigma_\nu^{(t)}. \tag{2.22}$$

D'autre part, la valeur de $L_\Lambda^{(t)}(X^{(t)})$ sur le programme de base $X_\nu^{(t)}$ ($\nu = 1, 2, \ldots, N_t^{(1)}$) ou sur le vecteur directeur $X_\nu^{(t)}$ d'une arête non bornée ($\nu = N_t^{(1)} + 1, \ldots, N_t$) est égale à

$$L_\Lambda^{(t)}(X_\nu^{(t)}) = (C^{(t)},\ X_\nu^{(t)}) - (\overline{\Lambda}^{(1)} A_t^{(0)},\ X_\nu^{(t)}).$$

En vertu de (2.13) et (2.14),

$$L_{\Lambda}^{(t)}(X_{\nu}^{(t)}) = \sigma_{\nu}^{(t)} - (\overline{\Lambda}^{(1)}, P_{\nu}^{(t)}). \tag{2.23}$$

En comparant (2.22) et (2.23), on obtient

$$L_{\Lambda}^{(t)}(X_{\nu}^{(t)}) = \varepsilon_{\nu}^{(t)}\lambda_{t}^{(2)} - \Delta_{\nu}^{(t)}. \tag{2.24}$$

Par conséquent, le critère $\Delta_{\nu}^{(t)}$ du vecteur des conditions $\overline{P}_{\nu}^{(t)}$ par rapport à la base du Z-problème se calcule d'après la relation

$$\Delta_{\nu}^{(t)} = -L_{\Lambda}^{(t)}(X_{\nu}^{(t)}) + \varepsilon_{\Lambda}^{(t)}\lambda_{t}^{(2)}. \tag{2.25}$$

La procédure de résolution de chaque $X_{\Lambda}^{(t)}$-problème peut se terminer par l'un des deux résultats suivants (1° ou 2°).

Le résultat 1° a lieu lorsque le $X_{\Lambda}^{(t)}$-problème est réalisable. Conformément aux considérations du paragraphe 1.3, le vecteur $\overline{P}_{\nu*}^{(t)}$ au critère $\Delta_{\nu*}^{(t)}$ minimal (parmi les vecteurs $\overline{P}_{\nu}^{(t)}$ et avec t fixé) est engendré par le programme de base optimal du $X_{\Lambda}^{(t)}$-problème. Dans ces conditions, d'après (2.25),

$$\Delta_{\nu*}^{(t)} = -L_{\Lambda}^{(t)}(X_{\nu*}^{(t)}) + \lambda_{t}^{(2)}. \tag{2.26}$$

Le résultat 2° a lieu quand la forme linéaire du $X_{\Lambda}^{(t)}$-problème est non bornée sur l'ensemble G_t de ses programmes. Comme nous l'avons vu au paragraphe 1.3, dans ce cas on met en évidence le vecteur directeur $X_{\nu*}^{(t)}$ d'une arête non bornée de l'ensemble G_t, qui engendre le vecteur $\overline{P}_{\nu*}^{(t)}$ au critère $\Delta_{\nu*}^{(t)}$ négatif. D'après (2.25),

$$\Delta_{\nu*}^{(t)} = -L_{\Lambda}^{(t)}(X_{\nu*}^{(t)}).$$

Si

$$\min_{t} \Delta_{\nu*}^{(t)} = 0, \tag{2.27}$$

le programme de base étudié du Z-problème en est une solution (du fait que tous les vecteurs des conditions $\overline{P}_{\nu}^{(t)}$ ont des critères non négatifs). Il est clair que (2.27) peut avoir lieu si la discussion de tout $X_{\Lambda}^{(t)}$-problème s'achève par le résultat 1°.

Mais si la condition (2.27) n'est pas respectée, le processus de résolution du Z-problème doit être poursuivi. On aura intérêt d'ajouter alors à la base successive du Z-problème le vecteur $\overline{P}_{\nu*}^{(k)}$ tel que

$$\Delta_{\nu*}^{(k)} = \min_{t} \Delta_{\nu*}^{(t)}. \tag{2.28}$$

Les composantes du vecteur $\overline{P}_{\nu*}^{(k)}$ se définissent suivant la formule

$$\overline{P}_{\nu*}^{(k)} = \begin{pmatrix} P_{\nu*}^{(k)} \\ \varepsilon_{\nu}^{(k)} e_k \end{pmatrix} = \begin{pmatrix} A_k^{(0)} X_{\nu*}^{(k)} \\ \varepsilon_{\nu}^{(k)} e_k \end{pmatrix}. \tag{2.29}$$

Le coefficient correspondant de la forme linéaire du Z-problème est égal à

$$\sigma_{\nu*}^{(k)} = (C^{(k)}, X_{\nu*}^{(k)}). \tag{2.30}$$

Tous les calculs ultérieurs liés à l'itération suivante de la procédure du Z-problème se font d'après les règles du deuxième algorithme de la méthode primale du simplexe.

Pour achever la description de la méthode de décomposition appliquée à la matrice diagonale par blocs $A^{(1)}$, il reste encore à faire certaines remarques sur la détermination du programme de base initial du Z-problème.

Le calcul de ce programme ne présente aucune difficulté quand toutes les contraintes du X-problème sont des inégalités aux deuxièmes membres non négatifs. Dans ce cas, il est rationnel d'introduire dans les conditions du Z-problème m variables supplémentaires $z_i^{s+1} \geqslant 0$ et de prendre comme base de départ la matrice

$$(\bar{P}_0^{(1)}, \bar{P}_0^{(2)}, \ldots, \bar{P}_0^{(s)}, \bar{P}_1^{(s+1)}, \bar{P}_2^{(s+1)}, \ldots, \bar{P}_m^{(s+1)} = \begin{Vmatrix} 0 & E_m \\ E_s & 0 \end{Vmatrix}. \tag{2.31}$$

E_k est ici la matrice unité d'ordre k. Les variables de base associées à la base initiale du Z-problème sont égales respectivement à

$$\underbrace{1, 1, \ldots, 1}_{s}, b_1^{(0)}, b_2^{(0)}, \ldots, b_m^{(0)},$$

où les $b_i^{(0)}$ sont les composantes du vecteur $B^{(0)}$.

La matrice inverse qui correspond à la base initiale s'écrit

$$\begin{Vmatrix} 0 & E_s \\ E_m & 0 \end{Vmatrix}.$$

En partant de la base initiale et du programme de base qui lui est associé on résout le Z-problème conformément à la procédure exposée plus haut.

Pour appliquer la méthode de décomposition à la résolution des problèmes à conditions-inégalités, il est bon de retenir les considérations qui suivent.

Les critères de sélection par rapport à la base courante des vecteurs des conditions $\bar{P}_i^{(s+1)}$ associés aux variables supplémentaires z_i^{s+1} se calculent d'après les relations

$$\Delta_i^{(s+1)} = (\bar{\Lambda}, \bar{P}_i^{(s+1)}) - \sigma_i^{(s+1)} = (\Lambda^{(1)}, P_i^{(s+1)}) - 0 = \\ = (\Lambda^{(1)}, e_i) = \lambda_i^{(1)}. \tag{2.32}$$

Si donc à une certaine itération de la résolution du Z-problème parmi les composantes du vecteur $\Lambda^{(1)}$ figure au moins une composante négative, il n'est plus besoin de construire à cette itération

des $X_\lambda^{(t)}$-problèmes. On introduit dans la base du Z-problème le vecteur unité des conditions supplémentaire associé à un critère négatif.

Si le X-problème est donné sous une forme canonique, la recherche du programme de base initial impose des calculs plus lourds. Il faut se servir dans ce cas de l'un quelconque des procédés décrits dans [87] (§ 6 du chapitre 7), où l'on trouve la description des méthodes de calcul du programme de base initial d'un problème de programmation linéaire. Voici *grosso modo* certains d'entre eux qu'on peut appliquer à la méthode de décomposition adaptée à une matrice diagonale par blocs $A^{(1)}$.

Soit $X_0^{(t)}$ $(t = 1, 2, \ldots, s)$ une collection des s programmes de base initiaux des $X^{(t)}$-problèmes. Ils permettent de construire s vecteurs des conditions du Z-problème

$$\bar{P}_0^{(t)} = \begin{pmatrix} A_t^{(0)} X_0^{(t)} \\ e_t \end{pmatrix}.$$

Introduisons m variables non négatives artificielles

$$Z^{(s+1)} = (z_1^{(s+1)}, z_2^{(s+1)}, \ldots, z_m^{(s+1)})$$

et énonçons le problème auxiliaire suivant.

Maximiser la forme linéaire

$$L_{s+1} = -\sum_{i=1}^{m} z_i^{(s+1)} \tag{2.33}$$

sous les conditions

$$\sum_{t=1}^{s} \bar{P}_0^{(t)} z_0^{(t)} + \sum_{t=1}^{s} \sum_{\nu=1}^{N_t} \bar{P}_\nu^{(t)} z_\nu^{(t)} + \bar{E} Z^{(s+1)} = B^{(0)}; \tag{2.34}$$

$$z_\nu^{(t)} \geqslant 0, \quad \nu = 0, 1, \ldots, N_t; \quad t = 1, 2, \ldots, s; \tag{2.35}$$

$$Z^{(s+1)} \geqslant 0. \tag{2.36}$$

Ici

$$\bar{E} = \begin{pmatrix} E_m \\ 0_{s \times m} \end{pmatrix}.$$

En supposant non négatives toutes les composantes du vecteur $\bar{B}^{(0)}$, on obtient une base de programme initial du problème auxiliaire, composées des vecteurs

$$\bar{P}_0^{(1)}, \bar{P}_0^{2}, \ldots, \bar{P}_0^{(s)}, \bar{P}_1^{(s+1)}, \ldots, \bar{P}_m^{(s+1)},$$

et les valeurs correspondantes des composantes de base de ce programme de base initial

$$z_0^{(t)} = 1, \quad t = 1, 2, \ldots, s,$$
$$z_i^{(s+1)} = b_i^{(0)}, \quad i = 1, 2, \ldots, m.$$

En partant de ce programme de base on résout le problème auxiliaire d'après les règles de la méthode de décomposition. Si la valeur optimale de la forme linéaire L_{s+1} est négative, le Z-problème, et par conséquent, le X-problème qui lui est équivalent, sont irréalisables. Si la valeur maximale de L_{s+1} sous les conditions (2.34)-(2.36) est nulle, la solution du problème auxiliaire peut être considérée comme programme de base initial du Z-problème. (Pour plus de détails, voir ce problème dans [87], chap. 8, § 6).

Le calcul du programme de départ et la résolution du Z-problème peuvent se faire sans diviser la procédure en deux étapes. A cette fin la M-méthode (cf. § 6, chapitre 7, [87]) doit être reformulée en termes du Z-problème. Toutefois, dans ce cas-là aussi, le nombre de variables artificielles du Z-problème est réduit jusqu'à m de façon analogue.

2.2. Voici un exemple pour illustrer les raisonnements du paragraphe précédent.

Soit la forme linéaire à maximiser

$$L(X) = 2x_1 + 2x_2 + 6x_3 - x_4 + 6x_6 + 8x_7 + x_8 - 9x_9 + 4x_{10}$$

sous les conditions

$$\begin{aligned}
-7x_1 + 8x_2 + 5x_3 - 2x_4 - 3x_5 + x_6 - 8x_7 - 4x_8 \qquad\qquad + 9x_{10} &\leqslant 4,\\
8x_1 + 4x_2 + x_3 + 2x_4 + 8x_5 + 5x_6 + 3x_7 + 9x_8 + 5x_9 + 8x_{10} &\leqslant 26,\\
3x_1 - 6x_2 + x_3 &\leqslant 5,\\
5x_1 - 3x_2 + 3x_3 &\leqslant 9,\\
5x_1 + 2x_2 + x_3 &\leqslant 8,\\
-7x_4 + 4x_5 + 9x_6 &\leqslant 4,\\
3x_4 - x_6 &\leqslant 3,\\
4x_7 + 3x_8 + x_9 - 7x_{10} &\leqslant 6,\\
7x_7 - 6x_9 + 3x_{10} &\leqslant 10,\\
x_j \geqslant 0,\ j = 1, 2, \ldots, 10.
\end{aligned}$$

Le Z-problème correspondant compte cinq conditions: deux inégalités, définies par les deux premières contraintes du X-problème, et une condition du type (2.11) par bloc.

En nous guidant sur les recommandations du paragraphe 2.1 [cf. (2.31)], nous pouvons prendre pour base initiale du Z-problème le système de vecteurs

$$(\overline{P}_0^{(1)}, \overline{P}_0^{(2)}, \overline{P}_0^{(3)}, \overline{P}_1^{(4)}, \overline{P}_2^{(4)}) = \begin{Vmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{Vmatrix},$$

et pour programme de base initial, le vecteur à composantes de base

$$(1, 1, 1, 4, 26).$$

Dans l'exemple considéré le vecteur nul $\overline{\Lambda}$ des critères des conditions du Z-problème correspond à la base initiale. C'est pourquoi les $X_{\Lambda}^{(t)}$-problèmes respectifs coïncident avec les $X^{(t)}$-problèmes. Il est évident que dans le cas considéré, G est un ensemble borné.

Il est commode de disposer les calculs qui reflètent le processus de résolution du Z-problème dans les tableaux recommandés pour le problème général de programmation linéaire lors de la description de l'algorithme de la méthode de décomposition. Pour les $X_{\Lambda}^{(t)}$-problèmes, on doit naturellement prévoir aussi une ou deux séries de tableaux suivant que la solution de ces problèmes s'obtient à partir du premier ou du deuxième algorithme de la méthode primale du simplexe.

Les coefficients des formes linéaires des $X_{\Lambda}^{(t)}$-problèmes répondant aux itérations successives du Z-problème sont portés dans le tableau II, où figurent, en outre, les programmes de base et les vecteurs directeurs des arêtes infinies du domaine de définition des $X_{\Lambda}^{(t)}$-problèmes, associés aux vecteurs introduits au cours de la résolution dans la base du Z-problème. La dernière ligne du tableau II donne le programme optimal du problème de départ.

Les programmes optimaux des $X_{\Lambda}^{(t)}$-problèmes ($t = 1$ et 2) qui correspondent à l'itération initiale de la procédure de résolution du Z-problème, contiennent les composantes de base suivantes:

pour le $X_{\Lambda}^{(1)}$-problème

$$x_1 = 1{,}667; \quad x_2 = 4{,}667; \quad x_{13} = 10{,}333;$$

pour le $X_{\Lambda}^{(2)}$-problème,

$$x_4 = 1{,}55; \quad x_6 = 1{,}65.$$

La solution du $X_{\Lambda}^{(3)}$-problème associé à l'itération initiale de la procédure de résolution du Z-problème est irréalisable. Sa forme linéaire est infinie sur l'ensemble des programmes du problème.

Les valeurs optimales des formes linéaires des $X_{\Lambda}^{(t)}$-problèmes ($t = 1$ et 2) sont respectivement égales à 31,333 et 8,350. D'après la formule (2.26), les critères des vecteurs des conditions du Z-problème définis par les solutions des $X_{\Lambda}^{(t)}$-problèmes ($t = 1, 2$) sont égaux à:

$$\Delta_{\nu*}^{(1)} = -31{,}333; \quad \Delta_{\nu*}^{(2)} = -8{,}350.$$

Le critère du vecteur des conditions du Z-problème associé à l'arête non bornée du domaine de définition du $X_{\Lambda}^{(3)}$-problème est égal à $\Delta^{(3)} = -3{,}333$. L'arête non bornée est issue du sommet correspondant au programme de base du $X_{\Lambda}^{(3)}$-problème,

$$X_1^3 = (0;\ 9{,}778;\ 0;\ 3{,}333).$$

Le programme de base $X_1^{(3)}$ détermine le vecteur des conditions du Z-problème dont le critère $\Delta_{\nu}^{(3)} = -23{,}111$.

Le plus faible de ces quatre critères correspond au $X_{\Lambda}^{(1)}$-problème. Cela signifie qu'il faut introduire dans la base du Z-problème un vecteur $\overline{P}_1^{(1)}$ dont les composantes se calculent d'après la formule (2.29). Dans notre cas,

$$\overline{P}_1^{(1)} = (36{,}667;\ 11{,}333;\ 1;\ 0;\ 0).$$

Le coefficient $\sigma_{\nu*}^{(k)}$ de la forme linéaire du Z-problème associé au vecteur des conditions $\overline{P}_1^{(1)}$ est égal, d'après la formule (2.30), à 31,333.

Les quantités $\lambda_i^{(2)}$, $\Delta_{\nu*}^{(f)}$, les composantes du vecteur des conditions $\overline{P}_{\nu*}^{(k)}$ ajouté à la base successive du Z-problème et le coefficient $\sigma_{\nu*}^{(k)}$ de la forme linéaire qui lui est associé sont inscrits pour chaque itération du Z-problème dans le tableau III.

Les règles de la méthode primale du simplexe impliquent l'élimination de la base du vecteur $\overline{P}_1^{(4)}$. En opérant selon ces mêmes règles avec la base $(\overline{P}_0^{(1)}, \overline{P}_0^{(2)}, \overline{P}_0^{(3)}, \overline{P}_1^{(1)}, \overline{P}_2^{(4)})$ nouvellement construite, on construit le programme de base suivant du Z-problème, on calcule les critères des conditions du problème par rapport à la nouvelle base et l'on résout les $X_\Lambda^{(f)}$-problèmes qu'elle a engendrés. Les critères des vecteurs des conditions du Z-problème définis par les programmes optimaux du $X_\Lambda^{(1)}$-problème et du $X_\Lambda^{(2)}$-problème sont respectivement égaux à $\Delta_{\nu*}^{(1)} = -12{,}836$ et $\Delta_{\nu*}^{(2)} = -9{,}589$. La solution du $X_\Lambda^{(3)}$-problème correspondant à la première itération de la procédure de résolution du Z-problème s'avère de nouveau irréalisable. Le vecteur $X_1^{(3)} = (0;\ 9{,}778;\ 0;\ 3{,}333)$ correspond au sommet d'où est issue l'arête non bornée du domaine de définition du $X_\Lambda^{(3)}$-problème. Le programme de base du $X_1^{(3)}$-problème détermine le vecteur des conditions $\overline{P}_1^{(3)}$ du Z-problème au critère $\Delta_1^{(3)} = -30{,}897$. Le vecteur directeur de l'arête non bornée du $X_\Lambda^{(3)}$-problème [vecteur $X^{(3)} = (0;\ 4{,}333;\ 1;\ 2)$] correspond au vecteur des c nditions du Z-problème au critère $\Delta^{(3)} = -2{,}764$. Le plus petit critère correspond au vecteur $\overline{P}_1^{(3)}$. C'est ce dernier qu'il est donc rationnel d'inclure dans la base suivante. Le vecteur à exclure de la base est $\overline{P}_2^{(4)}$.

En poursuivant des calculs analogues, on aboutit, au terme de la sixième itération de la résolution du Z-problème, à trois $X_\Lambda^{(f)}$-problèmes dont chacun correspond à une valeur nulle de $\Delta_{\nu*}^{(f)}$. D'après le critère d'optimalité, la conclusion obtenue signifie que le programme de base fourni par la sixième itération est solution du Z-problème. Les composantes de base de ce programme sont égales à

$$z_1^{(1)} = 0{,}431;\ z_2^{(1)} = 0{,}307;\ z_3^{(1)} = 0{,}262;\ z_1^{(2)} = 1;\ z_2^{(3)} = 1.$$

Les programmes de base respectifs $X_1^{(1)}$, $X_2^{(1)}$, $X_3^{(1)}$, $X_1^{(2)}$, $X_2^{(3)}$ des $X_\Lambda^{(f)}$-problèmes sont figurés dans le tableau II. D'après les relations (2.18), la solution du X-problème est donnée par la formule

$$X^* = (X_1^{(1)}z_1^{(1)} + X_2^{(1)}z_2^{(1)} + X_3^{(1)}z_3^{(1)},\ X_1^{(2)}z_1^{(2)},\ X_2^{(3)}z_2^{(3)}).$$

Les calculs donnent les valeurs suivantes des composantes non nulles du programme optimal du X-problème:

$$x_1 = 0{,}461;\ x_2 = 0{,}718;\ x_3 = 2{,}949;\ x_4 = 1{,}550;\ x_6 = 1{,}650;$$
$$x_7 = 1{,}429;\ x_8 = 0{,}0952.$$

Les première et troisième conditions du $X^{(1)}$-problème sont satisfaites par le programme optimal en tant qu'inégalités strictes. Les valeurs des variables supplémentaires (variables d'écart des première et troisième conditions) sont respectivement

$$x_{13} = 4{,}973;\quad x_{15} = 1{,}311.$$

Toute la procédure de résolution du problème est portée sur les tableaux 4.6 à 4.11. Le premier d'entre eux contient les tableaux principaux correspondant aux itérations successives de la résolution du Z-problème selon les règles du deuxième algorithme de la méthode primale du simplexe. Le tableau 4.7 contient les paramètres du bloc $A^{(0)}$, les coefficients c_j de la forme linéaire du X-problème

Tableau 4.62

I

Z-problème

	N	$\sigma_\nu^{(l)}$	B_z	e_0	e_1	e_2	e_3	e_4	e_5	$\overline{P}_{\nu*}^{(k)}$	θ	N_z
	1		$\overline{P}_0^{(1)}$	1			1			1	1	
	2		$\overline{P}_0^{(2)}$	1				1			—	
	3		$\overline{P}_0^{(3)}$	1					1		—	
←	4		$\overline{P}_1^{(4)}$	4	1					36,667	$\boxed{0,109}$	0
	5		$\overline{P}_2^{(4)}$	26		1				11,333	2,294	
—	6	—	—							−31,333	—	
	1		$\overline{P}_0^{(1)}$	0,891	−0,0273		1			0,248	3,585	
	2		$\overline{P}_0^{(2)}$	1				1			—	
	3		$\overline{P}_0^{(3)}$	1					1	1	1	1
→	4	31,333	$\overline{P}_1^{(1)}$	0,109	0,0273					−0,248	—	
←	5		$\overline{P}_2^{(4)}$	24,764	−0,309	1				117,483	$\boxed{0,211}$	
—	6	—	—	3,418	0,855					−30,897	—	

	1		$\overline{P}_0^{(1)}$	0,839	−0,0266	−0,00212	1			0,303	2,763	
	2		$\overline{P}_0^{(2)}$	1				1			—	
←	3		$\overline{P}_0^{(3)}$	0,789	0,00263	−0,00851			1	0,925	$\boxed{0,853}$	2
	4	31,333	$\overline{P}_1^{(1)}$	0,161	0,0266	0,00212				−0,303	—	
→	5	23,111	$\overline{P}_1^{(3)}$	0,211	−0,00263	0,00851				0,0748	2,816	
—	6	—	—	9,931	0,773	0,263				−19,303	—	
←	1		$\overline{P}_0^{(1)}$	0,580	−0,0275	0,00068	1		−0,328	1,228	$\boxed{0,472}$	
	2		$\overline{P}_0^{(2)}$	1				1			—	
→	3	11,524	$\overline{P}_2^{(3)}$	0,853	0,0284	−0,00920			1,081	−0,138	—	
	4	31,333	$\overline{P}_1^{(1)}$	0,420	0,0275	−0,00068			0,328	−0,228	—	3
	5	23,111	$\overline{P}_1^{(3)}$	0,147	−0,0284	0,00920			−0,0809	0,138	1,067	
—	6	—	—	26,398	0,828	0,0854			20,865	−11,558	—	
→	1	6	$\overline{P}_2^{(1)}$	0,472	−0,0224	0,00055	0,814		−0,267	0,0387	12,194	
	2		$\overline{P}_0^{(2)}$	1				1		1	1	
	3	11,524	$\overline{P}_2^{(3)}$	0,918	−0,00024	−0,00912	0,112		1,044	−0,103	—	4
	4	31,333	$\overline{P}_1^{(1)}$	0,528	0,0224	−0,00055	0,186		0,267	−0,0387	—	

Suite

	N	$\sigma_\nu^{(l)}$	B_z	e_0	e_1	e_2	e_3	e_4	e_5	$\overline{P}_{\nu^*}^{(k)}$	θ	N_z
←	5	23,111	$\overline{P}_1^{(3)}$	0,0819	0,00024	0,00912	−0,112		−0,0441	0,103	$\boxed{0,794}$	
—	6	—	—	31,852	0,570	0,0918	9,409		17,778	−8,134	—	
	1	6	$\overline{P}_2^{(1)}$	0,441	−0,0225	−0,00287	0,856		−0,251	0,511	0,864	
←	2		$\overline{P}_0^{(2)}$	0,206	−0,00231	−0,0884	1,087	1	0,427	0,787	$\boxed{0,262}$	
	3	11,524	$\overline{P}_2^{(3)}$	1				1	1		—	5
	4	31,333	$\overline{P}_1^{(1)}$	0,559	0,0225	0,00287	0,144		0,251	0,489	1,142	
→	5	8,350	$\overline{P}_1^{(2)}$	0,794	0,00231	0,0884	−1,087		−0,427	−0,787	—	
—	6	—	—	38,308	0,588	0,811	0,571		14,302	−6,171	—	
	1	6	$\overline{P}_2^{(1)}$	0,307	−0,0210	0,0545	0,151	−0,649	−0,528			
→	2	18	$\overline{P}_3^{(1)}$	0,262	−0,00294	−0,112	1,381	1,271	0,543			
	3	11,524	$\overline{P}_2^{(3)}$	1					1			6
	4	31,333	$\overline{P}_1^{(1)}$	0,431	0,0239	0,0579	−0,532	−0,622	−0,0153			
	5	8,350	$\overline{P}_1^{(2)}$	1				1				
—	6	—	—	39,926	0,570	0,117	9,094	7,844	17,654		—	

Tableau 4.7

II

	N	B	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}
	1	4	−7	8	5	−2	−3	1	−8	−4		9
	2	26	8	4	1	2	8	5	3	9	5	8
	3	C_x	2	2	6	−1		6	8	1	−9	4
$c_A^{(i)}$	1	—	7,982	−4,836	1,727	0,709	2,564	5,145	14,836	4,418	−9	−3,691
	2	—	5,309	−5,238	1,871	0,0205	0,216	3,912	13,397	1,726	−10,315	−5,063
	3	—	7,114	−4,967	1,774	0,486	1,801	4,745	14,369	3,544	−9,427	4,137
	4	—	5,253	−2,924	3,060	−0,044	0,975	4,972	12,281	2,452	−9,459	−1,860
	5	—	−0,368	−5,950	2,247	−1,445	−4,722	1,357	10,274	−3,944	−13,054	−7,782
	6	—	5,052	−3,031	3,031	0,094	0,771	4,843	12,210	2,224	−9,587	−2,071
$X_A^{(i)}$	1	$X_1^{(1)}$		1,667	4,667							
	2	$X_1^{(3)}$								9,778		3,333
	3	$X_2^{(3)}$							1,429	0,095		
	4	$X_2^{(1)}$	1,5		0,5							
	5	$X_1^{(2)}$				1,550		1,650				
	6	$X_3^{(1)}$			3							
		X^*	0,461	0,718	2,949	1,550		1,650	1,429	0,095		

Tableau 4.8

III

N_z	t	$\lambda_t^{(2)}$	$\Delta_{\nu *}^{(t)}$	$\sigma_{\nu *}^{(h)}$	$\overline{P}_{\nu *}^{(h)}$	Composantes du vecteur $\overline{P}_{\nu *}^{(h)}$				
0	1	0	$\boxed{-31,333}$	31,333	$\overline{P}_1^{(1)}$	36,667	11,333	1		
	2	0	−8,350							
	3	0	−23,111							
1	1	0	−12,836							
	2	0	−9,589							
	3	0	$\boxed{-30,897}$	23,111	$\overline{P}_1^{(3)}$	−9,111	114,667			1
2	1	0	−8,399							
	2	0	−6,486							
	3	0	$\boxed{-19,303}$	11,524	$\overline{P}_2^{(3)}$	−11,810	5,143			1

3	1	0	$\boxed{-11,558}$	6	$\overline{P}_2^{(1)}$	−8	12,5	1		
	2	0	−8,582							
	3	20,865	0							
4	1	9,409	0							
	2	0	$\boxed{-8,134}$	8,35	$\overline{P}_1^{(2)}$	−1,45	11,35		1	
	3	17,778	0							
5	1	0,571	$\boxed{-6,171}$	18	$\overline{P}_3^{(1)}$	15	3	1		
	2	0	−0,603							
	3	14,302	−0,375							
6	1	9,094	0							
	2	7,844	0							
	3	17,654	0							

Tableau 4.9

$X_\Lambda^{(1)}$-problème

$\boxed{0}$	—	c_j	—	—	2	2	6				—	—
	N	c_i	B	A_0	A_1	A_2	A_3	A_{13}	A_{14}	A_{15}	θ	N_Λ
	1		A_{13}	5	3	—6	1	1			5	0
←	2		A_{14}	9	5	—3	3		1		$\boxed{3}$	
	3		A_{15}	8	5	2	1			1	8	
—	4	—	—		—2	—2	$\boxed{-6}$				—	
	1		A_{13}	2	1,333	—5		1	—0,333		—	1
→	2	6	A_3	3	1,667	—1	1		0,333		—	
←	3		A_{15}	5	3,333	3			—0,333	1	$\boxed{1,667}$	
—	4	—	—	18	8	$\boxed{-8}$			2		—	
	1		A_{13}	10,333	6,889			1	—0,889	1,667		2
	2	6	A_3	4,667	2,778		1		0,222	0,333		
→	3	2	A_2	1,667	1,111	1			—0,111	0,333		
—	4	—	L_1^*	31,333	16,889				1,111	2,667	—	

	—	$c_j^{(\Lambda)}$	—	—	7,982	—4,836	1,727				—	—
$\boxed{1}$	N	$c_i^{(\Lambda)}$	B	A_0	A_1	A_2	A_3	A_{13}	A_{14}	A_{15}	θ	N_Λ
←	1		A_{13}	10,333	6,889			1	—0,889	1,667	$\boxed{1,5}$	
	2	1,727	A_3	4,667	2,778		1		0,222	0,333	1,68	0
	3	—4,836	A_2	1,667	1,111	1			—0,111	0,333	1,5	
—	4	—	—		$\boxed{-8,558}$				0,921	—1,036	—	
→	1	7,982	A_1	1,5	1			0,145	—0,129	0,242	6,2	
	2	1,727	A_3	0,5			1	—0,403	0,581	—0,339	—	
←	3	—4,836	A_2	0		1		—0,161	0,0323	0,0645	$\boxed{0}$	1
—	4	—	—	12,836				1,242	$\boxed{-0,183}$	1,034	—	
	1	7,982	A_1	1,5	1	4		—0,5		0,5		
	2	1,727	A_3	0,5		—18	1	2,5		—1,5		
→	3		A_{14}	0		31		—5	1	2		2
—	4	—	$L^*=$	12,836		5,673		0,327		1,4	—	

Suite

	—	$c_j^{(\Lambda)}$	—	—	5,309	—5,238	1,871				—	—
$\boxed{2}$	N	$c_i^{(\Lambda)}$	B	A_0	A_1	A_2	A_3	A_{13}	A_{14}	A_{15}	θ	N_Λ
	1	5,309	A_1	1,5	1	4		—0,5		0,5	0,375	0
	2	1,871	A_3	0,5		—18	1	2,5		—1,5	—	
←	3		A_{14}	0		31		—5	1	2	$\boxed{0}$	
—	4	—	—	8,899		$\boxed{-7,199}$		2,022		—0,152	—	
	1	5,309	A_1	1,5	1			0,145	—0,129	0,242		1
	2	1,871	A_3	0,5			1	—0,403	0,581	—0,339		
→	3	—5,238	A_2	0		1		—0,161	0,0323	0,0645		
—	4	—	$L_1^*=$	8,899				0,861	0,232	0,313	—	
	—	$c_j^{(\Lambda)}$	—	—	7,114	—4,967	1,774				—	—
$\boxed{3}$	N	$c_i^{(\Lambda)}$	B	A_0	A_1	A_2	A_3	A_{13}	A_{14}	A_{15}	0	N_Λ

	1	7,114	A_1	1,5	1			0,145	−0,129	0,242	—	0
	2	1,774	A_3	0,5			1	−0,403	0,581	−0,339	0,861	
←	3	−4,967	A_2	0		1		−0,161	0,0323	0,0645	$\boxed{0}$	
—	4	—	—	11,558				1,118	$\boxed{-0,0482}$	1,600	—	
	1	7,114	A_1	1,5	1	4		−0,5		0,5		1
	2	1,774	A_3	0,5		−18	1	2,5		−1,5		
→	3		A_{14}	0		31		−5	1	2		
—	4	—	$L_1^*=$	11,558		1,493		0,878		0,896	—	
$\boxed{4}$	—	$c_j^{(\Lambda)}$	—	—	5,253	−2,924	3,060				—	—
	N	$c_i^{(\Lambda)}$	B	A_0	A_1	A_2	A_3	A_{13}	A_{14}	A_{15}	0	N_Λ
	1	5,253	A_1	1,5	1	4		−0,5		0,5	0,375	0
	2	3,060	A_3	0,5		−18	1	2,5		−1,5	—	
←	3		A_{14}	0		31		−5	1	2	$\boxed{0}$	
—	4	—	—	9,409		$\boxed{-31,153}$		5,025		−1,964	—	

Suite

	1	5,253	A_1	1,5	1			0,145	−0,129	0,242		1
	2	3,060	A_3	0,5			1	−0,403	0,581	−0,339		
→	3	−2,924	A_2	0		1		−0,161	0,0323	0,0645		
—	4	—	$L_1^* =$	9,409				0	1,005	0,0456	—	
	—	$c_j^{(\Lambda)}$	—	—	−0,368	−5,950	2,247				—	—
$\boxed{5}$	N	$c_i^{(\Lambda)}$	B	A_0	A_1	A_2	A_3	A_{13}	A_{14}	A_{15}	0	N_Λ
	1	−0,368	A_1	1,5	1			0,145	−0,129	0,242	6,2	0
	2	2,247	A_3	0,5			1	−0,403	0,581	−0,339	—	
←	3	−5,950	A_2	0		1		−0,161	0,0323	0,0645	$\boxed{0}$	
—	4	—	—	0,571				0	1,161	$\boxed{-1,234}$	—	
←	1	−0,368	A_1	1,5	1	−3,75		0,75	−0,25		$\boxed{2}$	1
	2	2,247	A_3	0,5		5,25	1	−1,25	0,75		—	

→	3		A_{15}	0		15,5		−2,5	0,5	1	—	
—	4	—	—	0,571		19,131		$\boxed{-3{,}086}$	1,778		—	
→	1		A_{13}	2	1,333	−5		1	−0,333			
	2	2,247	A_3	3	1,667	−1	1		0,333			2
	3		A_{15}	5	3,333	3			−0,333	1		
—	4	—	$L_1^* =$	6,742	4,114	3,703			0,749		—	
$\boxed{6}$	—	$c_j^{(\Lambda)}$	—	—	5,052	−3,031	3,031			—	—	—
	N	$c_i^{(\Lambda)}$	B	A_0	A_1	A_2	A_3	A_{13}	A_{14}	A_{15}	θ	N_Λ
	1		A_{13}	2	1,333	−5		1	−0,333			
	2	3,031	A_3	3	1,667	−1	1		0,333			0
	3		A_{15}	5	3,333	3			−0,333	1		
—	4	—	$L_2^* =$	9,094	0	0			1,010		—	

Tableau 4.10

$X_A^{(2)}$-problème

$\boxed{0}$	—	$c_j^{(A)}$	—	—	—1		6			—	—
	N	$c_i^{(A)}$	B	A_0	A_4	A_5	A_6	A_{16}	A_{17}	0	N_A
←	1		A_{16}	4	—7	4	9	1		0,444	0
	2		A_{17}	3	3		—1		1	—	
—	3	—	—		1		$\boxed{-6}$			—	
→	1	6	A_6	0,444	—0,778	0,444	1	0,111		—	1
←	2		A_{17}	3,444	2,222	0,444		0,111	1	1,55	
—	3	—	—	2,667	$\boxed{-3,667}$	2,667		0,667		—	
	1	6	A_6	1,65		0,6	1	0,15	0,35		2
→	2	—1	A_4	1,55	1	0,2		0,05	0,45		
—	3	—	$L_2^* =$	8,35		3,4		0,85	1,65	—	

1	—	$c_j^{(\Lambda)}$	—	—	0,709	2,564	5,145			—	—
	N	$c_i^{(\Lambda)}$	B	A_0	A_4	A_5	A_6	A_{16}	A_{17}	θ	N_Λ
	1	5,145	A_6	1,65		0,6	1	0,15	0,35		0
	2	0,709	A_4	1,55	1	0,2		0,05	0,45		
—	3	—	$L_2^* =$	9,589		0,665		0,807	2,120	—	
2	—	$c_j^{(\Lambda)}$	—	—	0,0205	0,216	3,912			—	—
	N	$c_i^{(\Lambda)}$	B	A_0	A_4	A_5	A_6	A_{16}	A_{17}	θ	N_Λ
	1	3,912	A_6	1,65		0,6	1	0,15	0,35		0
	2	0,0205	A_4	1,55	1	0,2		0,05	0,45		
—	3	—	$L_2^* =$	6,486		2,135		0,588	1,378	—	

Suite

3	—	$c_j^{(\Lambda)}$	—	—	0,486	1,801	4,745			—	—
	N	$c_i^{(\Lambda)}$	B	A_0	A_4	A_5	A_6	A_{16}	A_{17}	θ	N_Λ
	1	4,745	A_6	1,65		0,6	1	0,15	0,35		0
	2	0,486	A_4	1,55	1	0,2		0,05	0,45		
—	3	—	$L_2^* =$	8,582		1,143		0,736	1,879		
4	—	$c_j^{(\Lambda)}$	—	—	—0,0444	0,975	4,971			—	—
	N	$c_i^{(\Lambda)}$	B	A_0	A_4	A_5	A_6	A_{16}	A_{17}	θ	N_Λ
	1	4,971	A_6	1,65		0,6	1	0,15	0,35		0
	2	—0,0444	A_4	1,55	1	0,2		0,05	0,45		
—	3	—	$L_2^* =$	8,134		2		0,744	1,720	—	

5	—	$c_j^{(\Lambda)}$	—	—	—1,445	—4,722	1,357			—	—
	N	$c_i^{(\Lambda)}$	B	A_0	A_4	A_5	A_6	A_{16}	A_{17}	θ	N_Λ
	1	1,357	A_6	1,65		0,6	1	0,15	0,35	4,714	0
←	2	—1,445	A_4	1,55	1	0,2		0,05	0,45	$\boxed{-3,444}$	
—	3	—	—	0		5,247		0,131	$\boxed{-0,175}$	—	
	1	1,357	A_6	0,444	—0,778	0,444	1	0,111			1
→	2		A_{17}	3,444	2,222	0,444		0,111	1		
—	3	—	$L_3^* =$	0,603	0,389	5,325		1,508		—	
6	—	$c_j^{(\Lambda)}$	—	—	—0,0943	0,771	4,843			—	—
	N	$c_i^{(\Lambda)}$	B	A_0	A_4	A_5	A_6	A_{16}	A_{17}	0	N_Λ
	1	4,843	A_6	0,444	—0,778	0,444	1	0,111		—	0
←	2		A_{17}	3,444	2,222	0,444		0,111	1	$\boxed{1,55}$	
—	3	—	—	2,152	$\boxed{-3,672}$	1,381		0,538		—	
	1	4,843	A_6	1,65		0,6	1	0,15	0,35		1
→	2	—0,0943	A_4	1,55	1	0,2		0,05	0,45		
—	3	—	$L_3^* =$	7,844		19,038		0,722	1,653	—	

Tableau 4.11

$X_A^{(3)}$-problème

$\boxed{0}$	—	c_j	—	—	8	1	—9	4			—	—
	N	c_i	B	A_0	A_7	A_8	A_9	A_{10}	A_{18}	A_{19}	θ	N_A
	1		A_{18}	6	4	3	1	—7	1		1,5	
←	2		A_{19}	10	7		—6	3		1	$\boxed{1,429}$	0
—	3	—	—		$\boxed{-3}$	—1	9	—4			—	
←	1		A_{18}	0,286		3	4,429	—8,714	1	—0,571	$\boxed{0,0952}$	
→	2	8	A_7	1,429	1		—0,857	0,429		0,143	—	1
—	3	—	—	11,429		$\boxed{-1}$	2,143	—0,571		1,143	—	
→	1	1	A_8	0,0952		1	1,476	—2,905	0,333	—0,190	—	
←	2	8	A_7	1,429	1		—0,857	0,429		0,143	$\boxed{3,333}$	2

—	3	—	—	11,524			3,619	$\boxed{-3,476}$	0,333	0,952	—	
	1	1	A_8	9,778	6,778	1	−4,333		0,333	0,778	—	
→	2	4	A_{10}	3,333	2,333		−2	1		0,333	—	3
—	3	—	$L_3^*=$	23,111	8,111		$\boxed{-3,333}$		0,333	2,111	—	
$\boxed{1}$	—	$c_j^{(\Lambda)}$	—	—	14,836	4,418	−9	−3,691			—	—
	N	$c_i^{(\Lambda)}$	B	A_0	A_7	A_8	A_9	A_{10}	A_{18}	A_{19}	0	N_Λ
	1	4,418	A_8	9,778	6,778	1	−4,333		0,333	0,778	—	
	2	−3,691	A_{10}	3,333	2,333		−2	1		0,333	—	0
—	3	—	$L_3^*=$	30,897	6,497		$\boxed{-2,764}$		1,473	2,206	—	
$\boxed{2}$	—	$c_j^{(\Lambda)}$	—	—	13,397	1,726	−10,315	−5,063			—	—
	N	$c_i^{(\Lambda)}$	B	A_0	A_7	A_8	A_9	A_{10}	A_{18}	A_{19}	0	N_Λ

Suite

	1	1,726	A_8	9,778	6,778	1	−4,333		0,333	0,778	1,443	
←	2	−5,063	A_{10}	3,333	2,333		−2	1		0,333	$\boxed{1,429}$	0
—	3	—	—		$\boxed{-13,512}$		12,962		0,575	−0,345	—	
	1	1,726	A_8	0,0952		1	1,476	−2,905	0,333	−0,190		
→	2	13,397	A_7	1,429	1		−0,857	0,429		0,143		1
—	3	—	$L_3^* =$	19,303			1,380	5,791	0,575	1,585	—	
	—	$c_j^{(\Lambda)}$	—	—	14,369	3,544	−9,427	−4,136			—	—
$\boxed{3}$	N	$c_i^{(\Lambda)}$	B	A_0	A_7	A_8	A_9	A_{10}	A_{18}	A_{19}	θ	N_Λ
	1	3,544	A_8	0,0952		1	1,476	−2,905	0,333	−0,190		
	2	14,369	A_7	1,429	1		−0,857	0,429		0,143		0
—	3	—	$L_3^* =$	20,865			2,342	0	1,181	1,378	—	

	—	$c_j^{(\Lambda)}$	—	—	12,281	2,452	—9,459	—1,860			—	—
$\boxed{4}$	N	$c_i^{(\Lambda)}$	B	A_0	A_0	A_8	A_9	A_{10}	A_{18}	A_{19}	0	N_Λ
	1	2,452	A_8	0,0952		1	1,476	—2,905	0,333	—0,190		
	2	12,281	A_7	1,429	1		—0,857	0,429		0,143		0
	3	—	$L_3^*=$	17,778			2,552	0	0,818	1,290	—	
	—	$c_j^{(\Lambda)}$	—	—	10,274	—3,944	—13,054	—7,782			—	—
$\boxed{5}$	N	$c_i^{(\Lambda)}$	B	A_0	A_7	A_8	A_9	A_{10}	A_{18}	A_{19}	0	N_Λ
←	1	—3,944	A_8	0,0952		1	1,476	—2,905	0,333	—0,190	$\boxed{0,0645}$	
	2	10,274	A_7	1,429	1		—0,857	0,429		0,143	—	0
—	3	—	—	14,302			$\boxed{-1,575}$	23,642	—1,315	2,219	—	

Suite

⇄	1	−13,054	A_9	0,0645		0,667	1	−1,968	0,226	−0,129	$\boxed{0,286}$	
	2	10,274	A_7	1,484	1	0,581		−1,258	0,194	0,0323	7,333	1
—	3	—	—	14,403		1,067		20,544	$\boxed{-0,959}$	2,016	—	
→	1		A_{18}	0,286		3	4,429	−8,714	1	−0,571		
	2	10,274	A_7	1,429	1		−0,857	0,429		0,143		2
—	3	—	$L_3^* =$	14,677		3,944	4,248	12,185		1,428	—	
$\boxed{6}$	—	$c_j^{(\Lambda)}$	—	—	12,210	2,224	−9,587	−2,071			—	—
	N	$c_i^{(\Lambda)}$	B	A_0	A_7	A_8	A_9	A_{10}	A_{18}	A_{19}	0	N_Λ
←	1		A_{18}	0,286		3	4,429	−8,714	1	−0,571	$\boxed{0,0952}$	
	2	12,210	A_7	1,429	1		−0,857	0,429		0,143	—	0
—	3	—	—	17,442		$\boxed{-2,224}$	−0,878	7,304		1,744	—	
→	1	2,224	A_8	0,0952		1	1,476	−2,905	0,333	−0,190		
	2	12,210	A_7	1,429	1		−0,857	0,429		0,143		1
—	3	—	$L_3^* =$	17,654			2,405	0,843	0,741	1,321	—	

et les coefficients $c_\Lambda^{(t)}$ des formes linéaires des $X_\Lambda^{(t)}$-problèmes engendrés par les bases successives du Z-problème.

Dans le tableau 4.7 figurent également les programmes optimaux des $X_\Lambda^{(t)}$-problèmes qui engendrent les vecteurs $\overline{P}_\nu^{(t)}$ ajoutés à la base successive des Z-problèmes ainsi que la solution du problème initial. Le tableau 4.8 réunit les valeurs des paramètres $\lambda_t^{(2)}$, L_t^* et $\Delta_{\nu*}^{(t)}$, qui permettent de discuter l'optimalité du programme de base du Z-problème, les composantes des vecteurs des conditions $\overline{P}_{\nu*}^{(k)}$ inclus dans la base suivante, et les valeurs correspondantes $\sigma_{\nu*}^{(k)}$ des coefficients de la forme linéaire du Z-problème.

Enfin, les trois derniers tableaux (4.9, 4.10 et 4.11) fixent la procédure de résolution des suites de $X_\Lambda^{(t)}$-problèmes ($t = 1, 2, 3$) à l'aide du premier algorithme de la méthode primale du simplexe. Remarquons que, comme d'habitude, le programme optimal du $X_\Lambda^{(t)}$-problème correspondant à la l-ième étape de la résolution du Z-problème est utilisé comme programme de base initial pour résoudre le $X_\Lambda^{(t)}$-problème engendré par la $(l + 1)$-ième itération du Z-problème.

2.3. Considérons certaines modifications de la méthode de décomposition. Pour simplifier l'exposé, supposons que l'ensemble G soit borné.

Lors de la description de la méthode de décomposition, on avait proposé d'introduire dans la base successive du Z-problème le vecteur des conditions associé au critère négatif

$$\Delta_{\nu*}^{(k)} = \min_t \Delta_{\nu*}^{(t)}.$$

Le volume de calculs peut parfois être réduit en supprimant le calcul et la comparaison entre eux de tous les critères $\Delta_{\nu*}^{(t)}$ et en ajoutant à la base le premier des vecteurs au critère négatif. On sous-entend ici par premier vecteur le vecteur à $\Delta_{\nu*}^{(t)} < 0$ qui correspond au plus petit numéro t. Dans ces conditions, la nécessité de résoudre les $X_\Lambda^{(t)}$-problèmes suivants (à numéros t plus grands) disparaît. Autrement dit, on a intérêt à ne résoudre les $X_\Lambda^{(t)}$-problèmes que jusqu'à l'apparition d'un problème dont la solution correspondra à un vecteur des conditions du Z-problème ayant un critère négatif.

On peut aller encore plus loin. Nous avons parlé jusqu'à présent de l'inclusion dans la base successive du Z-problème du vecteur des conditions associé au programme optimal de l'un des $X_\Lambda^{(t)}$-problèmes. Or, à chacun des programmes de base $X_\nu^{(t)}$ de n'importe quel des $X_\Lambda^{(t)}$-problèmes correspond un certain vecteur des conditions du Z-problème. On voit de la formule (2.25) que le critère de ce vecteur s'écrit

$$\Delta_\nu^{(t)} = \lambda_t^{(2)} - L_\Lambda^{(t)}(X_\nu^{(t)}).$$

Pour passer à une nouvelle base du Z-problème, nul besoin n'est en règle générale d'obtenir le programme optimal du $X_\Lambda^{(t)}$-problème. Il suffit d'obtenir au cours de l'amélioration du programme du

$X_\Lambda^{(t)}$-problème le programme de base $X_\nu^{(t)}$ tel que

$$L_\Lambda^{(t)}\,(X_\nu^{(t)}) > \lambda_t^{(2)}.$$

Le critère du vecteur des conditions correspondant $\overline{P}_\nu^{(t)}$ du Z-problème sera négatif. L'introduction du vecteur $\overline{P}_\nu^{(1)}$ dans la base successive permettra d'améliorer le programme du Z-problème et de s'approcher de sa solution.

Ainsi, en appliquant la modification suivante de l'ordre des calculs on peut espérer que le volume de calculs de la méthode de décomposition sera réduit.

Chaque itération du Z-problème fournit un vecteur $\Lambda^{(1)}$ et les nombres $\lambda_t^{(2)}$. Composons le $X_\Lambda^{(1)}$-problème et calculons à chaque étape de sa résolution la valeur de la forme linéaire L_1. Lorsque $L_1\,(X_\nu^{(1)})$ s'accroît et dépasse la valeur $\lambda_1^{(2)}$ d'une certaine valeur, modifiable en cours de résolution, on met fin à la résolution du $X_\Lambda^{(1)}$-problème pour calculer le vecteur

$$\overline{P}_\nu^{(1)} = \begin{pmatrix} A_1^{(0)} X_\nu^{(1)} \\ e_1 \end{pmatrix},$$

l'introduire dans la base et passer à l'itération suivante du Z-problème.

Si la valeur de L_1 ne dépasse $\lambda_1^{(2)}$ de la valeur susmentionnée sur aucun des programmes de base du $X_\Lambda^{(1)}$-problème, y compris sur le programme de base optimal $X_{\nu*}^{(1)}$, on passe à la résolution du $X_\Lambda^{(2)}$-problème. En comparant, au cours de la résolution du $X_\Lambda^{(2)}$-problème, les valeurs de la forme linéaire L_2 sur les programmes de base successifs $X_\nu^{(2)}$ avec la valeur $\lambda_2^{(2)}$, on détermine s'il est possible de passer à l'itération suivante du Z-problème ou s'il faut résoudre le $X_\Lambda^{(3)}$-problème par blocs successif. Une telle voie de résolution du problème réduit le nombre de $X_\Lambda^{(t)}$-problèmes à explorer et permet dans plusieurs cas de ne pas pousser leur résolution jusqu'au bout.

Il ne semble guère utile, en général, de commencer chaque itération du Z-problème par l'examen du $X_\Lambda^{(t)}$-problème. Dans de tels cas la préférence irait plutôt aux $X_\Lambda^{(t)}$-problèmes à petits numéros de t qu'aux autres $X_\Lambda^{(t)}$-problèmes. Dans le cas général, il est apparemment plus rationnel de s'en tenir à un ordre cyclique de discussion des $X_\Lambda^{(t)}$-problèmes. Si le passage à la l-ième itération a eu lieu après résolution du $X_\Lambda^{(t_0)}$-problème, il est naturel de commencer la l-ième itération de la résolution du Z-problème par l'examen du $X_\Lambda^{(t_0+1)}$-problème. Si la résolution du $X_\Lambda^{(s)}$-problème n'offre pas de raison pour passer à la $(l+1)$-ième itération du Z-problème, il est bon de revenir au $X_\Lambda^{(t)}$-problème, etc.

2.4. Au cours de la résolution du Z-problème, on peut à chaque itération introduire dans la base non pas un seul vecteur à critère $\Delta_{\nu*}^{(k)}$ minimal, mais une série de vecteurs à valeurs négatives des critères $\Delta_{\nu*}^{(l)}$. L'observation des données ci-dessus permet d'introduire dans la base successive du Z-problème, en plus des vecteurs associés aux programmes optimaux des $X_\Lambda^{(l)}$-problèmes, aussi les vecteurs correspondant aux programmes de base $X_\nu^{(l)}$ de ces problèmes. Il importe seulement que les vecteurs des conditions $\overline{P}_\nu^{(l)}$ définis par les programmes de base $X_\nu^{(l)}$ aient par rapport à la base courante des critères négatifs $\Delta_\nu^{(l)}$. Les modifications à apporter dans l'algorithme de résolution du Z-problème se déduisent de la description de la modalité de la méthode primale du simplexe donnée dans [87] (§ 2 du chapitre 12).

Le procédé indiqué permet de réaliser simultanément plusieurs itérations du Z-problème en utilisant les résultats de la résolution d'une seule série de $X_\Lambda^{(l)}$-problèmes (associés au vecteur fixe Λ). Si la discussion des $X_\Lambda^{(l)}$-problèmes impose un travail assez considérable, en procédant ainsi on aboutit naturellement plus vite à la solution recherchée.

Voici la modification de l'algorithme de la méthode de décomposition qui semble présenter quelque intérêt. Supposons connus le programme de base initial du Z-problème et les critères des conditions du problème qui lui sont associés (vecteur $\Lambda^{(1)}$ et nombres $\lambda_l^{(2)}$). La résolution successive des $X_\Lambda^{(l)}$-problèmes engendrés par le vecteur $\Lambda^{(1)}$ permet d'écrire leurs programmes de base tels que

$$\Delta_\nu^{(l)} = \lambda_l^{(2)} - L_l\,(X_\nu^{(l)}) < 0.$$

A tous ces programmes de base correspondent des vecteurs des conditions du Z-problème dont l'inclusion dans la base améliore le programme de ce problème (augmente la valeur de la forme linéaire du Z-problème). Les composantes des vecteurs $\overline{P}_\nu^{(l)}$ et les valeurs des coefficients $\sigma_\nu^{(l)}$ se calculent d'après les formules (2.14), (2.16) et (2.13). Le calcul des vecteurs $\overline{P}_\nu^{(l)}$ se poursuit tant qu'ils ne saturent, avec les vecteurs de la base courante, la mémoire à accès immédiat de la machine. Appelons Z-problème tronqué le problème ainsi obtenu. Le Z-problème tronqué diffère du Z-problème par le fait qu'on y considère seulement les variables $z_\nu^{(l)}$ correspondant à la collection étudiée de vecteurs $\overline{P}_\nu^{(l)}$; les autres variables sont supposées d'avance nulles.

Le Z-problème tronqué se résout selon les règles générales de la méthode du simplexe. On ajoute à la base successive soit un seul, soit plusieurs vecteurs des conditions, en fonction des traits particuliers propres au problème. Les transformations élémentaires des

programmes de base se poursuivent tant que l'on obtienne le programme optimal du Z-problème tronqué.

Après chaque itération du Z-problème tronqué il n'est pas nécessaire de résoudre les $X_\Lambda^{(t)}$-problèmes. Ce n'est qu'après avoir obtenu le programme optimal du Z-problème tronqué qu'il est avantageux de construire une nouvelle série de $X_\Lambda^{(t)}$-problèmes, en se servant du vecteur $\Lambda^{(1)}$ des critères des m premières conditions du problème. En comparant les valeurs des formes linéaires L_t sur les programmes de base $X_\nu^{(t)}$ des problèmes décomposés en blocs avec les critères $\lambda_t^{(2)}$ des s dernières conditions du Z-problème tronqué, on définit les signes des critères $\Delta_\nu^{(t)}$ dès vecteurs $\overline{P}_\nu^{(t)}$ associés aux programmes de base $X_\nu^{(t)}$. Les vecteurs des conditions $\overline{P}_\nu^{(t)}$ aux critères négatifs forment avec les vecteurs de la base du programme optimal du problème tronqué précédent le Z-problème tronqué successif. Il est logique de choisir comme programme de base initial du nouveau problème tronqué le programme optimal du problème précédent.

L'application conséquente de l'artifice décrit dans ce qui précède permettra d'utiliser rationnellement la mémoire à accès immédiat de la machine et de réduire nettement le nombre d'itérations du Z-problème après lesquelles il faut résoudre les $X_\Lambda^{(t)}$-problèmes décomposés en blocs.

2.5. Lorsque le nombre de problèmes décomposés en blocs est important, on a intérêt à réunir les groupes de blocs en blocs distincts. On diminue ainsi le nombre de contraintes du type (2.11) dans le Z-problème. La résolution du $X_\Lambda^{(t)}$-problème correspondant à un groupe réuni se ramène naturellement à l'analyse d'une série de problèmes indépendants associés à chaque bloc. Si, par exemple, deux blocs (t_1 et t_2) sont réunis en un seul bloc (t_3), alors, dans le $X_\Lambda^{(t_3)}$-problème correspondant au bloc réuni il faut calculer le vecteur $X^{(t_3)} = (X^{(t_1)}, X^{(t_2)})$ de dimension $(n_{t_1} + n_{t_2})$ sur lequel s'obtient le maximum de la forme linéaire

$$(C^{(t_1)} - \Lambda^{(1)} A_0^{(t_1)},\ X^{(t_1)}) + (C^{(t_2)} - \Lambda^{(1)} A_0^{(t_2)},\ X^{(t_2)})$$

sous les conditions

$$A^{(t_1)} X^{(t_1)} = B^{(t_1)},$$
$$A^{(t_2)} X^{(t_2)} = B^{(t_2)},$$
$$(X^{(t_1)},\ X^{(t_2)}) \geqslant 0.$$

Il est clair que le calcul du programme optimal $X^{(t_3)}$ se ramène à la résolution de deux problèmes indépendants de programmation linéaire, de dimensions $m_{t_1} \times n_{t_1}$ et $m_{t_2} \times n_{t_2}$.

Lors de la réunion des blocs, on réunit les vecteurs des conditions du Z-problème qu'ils engendrent et dont les critères définissent si le programme est optimal ou s'il faut l'améliorer. Le vecteur des

conditions $\overline{P}_{\nu*}^{(t_3)}$ se calcule d'après la formule

$$\overline{P}_{\nu*}^{(t_3)} = \begin{pmatrix} P_{\nu*}^{(t_3)} \\ e_{t_3} \end{pmatrix} = \begin{pmatrix} A_{t_1}^{(0)} X_{\nu*}^{(t_1)} + A_{t_2}^{(0)} X_{\nu*}^{(t_2)} \\ e_{t_3} \end{pmatrix}.$$

Dans le cas limite, on peut réunir tous les blocs en un seul et réduire ainsi à $m+1$ le nombre de contraintes de Z-problème. Si l'on adopte le schéma de résolution du problème exposé au paragraphe 2.1, d'après lequel on inclut dans la base du Z-problème le vecteur au critère $\Delta_{\nu*}^{(k)} = \min\limits_{t} \Delta_{\nu*}^{(t)}$, les avantages éventuels de la diminution du nombre de blocs n'interviennent alors que dans une itération isolée du Z-problème. Or, nous avons déjà dit au paragraphe 2.3, qu'il est parfois utile d'inclure dans la base du Z-problème le premier vecteur $\overline{P}_{\nu}^{(t)}$ dont le critère serait négatif. Dans ce cas, le nombre de $X_{\Lambda}^{(t)}$-problèmes à soumettre à l'exploration diminue, et de plus, la résolution ne doit pas forcément être menée jusqu'à la fin pour chacun d'entre eux. Lors de la réunion des blocs, le choix du vecteur à inclure dans la base successive du Z-problème se fait, en général, après résolution d'un grand nombre de $X_{\Lambda}^{(t)}$-problèmes. Ceci peut conduire à un accroissement du volume global de travail exigé par la résolution du problème.

En outre, à mesure que diminue le nombre de blocs, le nombre total de programmes de base des $X^{(t)}$-problèmes augmente de même, par conséquent, que le nombre de variables du Z-problème. C'en est de plus un argument contre la réunion des blocs.

Des recommandations bien fondées sur les conditions rationnelles dans l'utilisation de telles ou telles modalités de la méthode de décomposition ne peuvent être faites qu'après l'accumulation d'une certaine expérience des calculs. A l'heure actuelle, cette expérience n'est pas encore acquise.

2.6. Il semble rationnel d'examiner la possibilité suivante d'extension du domaine d'application des modifications considérées de la méthode de décomposition. Jusqu'à présent, la forme générale du problème étudié par la méthode de décomposition était définie par les relations (2.1)-(2.4). Il est clair que toutes les considérations exposées peuvent être reformulées pour un problème à matrice des conditions du type

$$\overline{A} = \begin{Vmatrix} c^{(0)} & c^{(1)} & c^{(2)} & \dots & c^{(s)} \\ A_1^{(0)} & A^{(1)} & 0 & \dots & 0 \\ A_2^{(0)} & 0 & A^{(2)} & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ A_s^{(0)} & 0 & 0 & \dots & A^{(s)} \end{Vmatrix}.$$

Le problème dual au problème précédent est du type (2.1)-(2.4). C'est pourquoi l'application des principes de la méthode duale du

simplexe permet d'étendre également la procédure de la méthode de décomposition aux problèmes dans lesquels l'« enchaînement » s'effectue non pas aux dépens des lignes communes, mais aux dépens des colonnes communes de la matrice des conditions.

On peut aller encore plus loin. Considérons le problème dans lequel l'« enchaînement » s'effectue aussi bien aux dépens des lignes communes que des colonnes communes. Décomposons les conditions du problème en deux blocs. Attribuons au premier bloc les lignes d'« enchaînement », et au deuxième, les conditions restantes. En appliquant la méthode de décomposition à la résolution de ce problème, on remarque que la matrice des conditions du X_Λ-problème a une structure particulière : l'« enchaînement » des blocs (variables) s'y effectue aux dépens des colonnes. Ainsi, pour résoudre le X_Λ-problème on peut utiliser le principe de décomposition énoncé en termes de la méthode duale du simplexe. En décomposant les conditions et les variables en un plus grand nombre de groupes et en répétant à plusieurs reprises la procédure choisie, on peut ramener la résolution de n'importe quel problème de programmation linéaire de grande taille (y compris le problème de type général) à la résolution d'une série de problèmes dont le nombre de contraintes et de variables ne dépasse pas les valeurs données d'avance. Toutefois, la réalisation de cette proposition est liée à des calculs nombreux et lourds.

§ 3. Méthode duale de discussion des problèmes de programmation par blocs

3.1. Dans les paragraphes précédents nous avons examiné en détail l'une des approches possibles de résolution des problèmes décomposés en blocs : la méthode de décomposition de Dantzig et Wolfe.

Dans le présent paragraphe, nous allons donner une formulation duale du problème de programmation par blocs, dont le sens géométrique est immédiat.

Non seulement cette formulation permet de considérer d'une optique légèrement différente la méthode de décomposition et de donner son interprétation géométrique, mais encore, ce qui est particulièrement important, de mettre en évidence une méthode duale suffisamment générale pour discuter les problèmes décomposés en blocs.

Considérons le problème du § 1 :

$$(C,\ X) \to \max, \tag{3.1}$$

$$A^{(0)}X = B^{(0)}, \tag{3.2}$$

$$A^{(1)}X = B^{(1)} \tag{3.3}$$

$$X \geqslant 0. \tag{3.4}$$

$A^{(0)}$ et $A^{(1)}$ sont ici les matrices de dimensions $m \times n$ et $m_1 \times n$ respectivement;

C et X, les vecteurs à n composantes;

$B^{(0)}$ et $B^{(1)}$, les vecteurs de dimensions m et m_1 respectivement.

Comme précédemment, nous entendrons par G l'ensemble convexe de points X, défini par les conditions (3.3), (3.4).

Le problème (3.1)-(3.4) engendre le Z-problème:

$$\sum_{\nu=1}^{N} \sigma_\nu z_\nu \to \max, \tag{3.5}$$

$$\sum_{\nu=1}^{N} \bar{P}_\nu z_\nu = \bar{B}^{(0)}, \tag{3.6}$$

$$z_\nu \geqslant 0, \quad \nu = 1, 2, \ldots, N. \tag{3.7}$$

Rappelons que

$$\bar{P}_\nu = \begin{pmatrix} P_\nu \\ \varepsilon_\nu \end{pmatrix}, \quad P_\nu = A^{(0)} X_\nu, \quad \bar{B}^{(0)} = \begin{pmatrix} B^{(0)} \\ 1 \end{pmatrix},$$

$$\varepsilon_\nu = \begin{cases} 1 \text{ si } X_\nu \text{ est un sommet du polyèdre de l'ensemble } G; \\ 0 \text{ si } X_\nu \text{ est le vecteur directeur d'une arête infinie de l'ensemble } G; \end{cases}$$

$\sigma_\nu = (C, X_\nu)$; X_ν ($\nu = 1, 2, \ldots, N$) est une collection complète des sommets et des vecteurs directeurs des arêtes infinies de l'ensemble G.

Numérotons, comme auparavant, les ensembles X_ν de façon que les N_1 premiers des vecteurs X_ν soient associés aux sommets de l'ensemble G et les $N - N_1$ autres, aux arêtes infinies de cet ensemble. (Si G est un ensemble borné, $N - N_1 = 0$).

Introduisons également dans la discussion le $\tilde{Z}$-problème dual par rapport au Z-problème (3.5)-(3.7):

$$(\bar{B}^{(0)}, \bar{\Lambda}) = (B^{(0)}, \Lambda) + \lambda \to \min, \tag{3.8}$$

$$(P_\nu, \Lambda) + \lambda \geqslant \sigma_\nu, \quad \nu = 1, 2, \ldots, N_1, \tag{3.9}$$

$$(P_\nu, \Lambda) \geqslant \sigma_\nu, \quad \nu = N_1 + 1, \ldots, N, \tag{3.10}$$

où

$$\bar{\Lambda} = (\Lambda, \lambda); \quad \Lambda = (\lambda_1, \lambda_2, \ldots, \lambda_m).$$

Puisque la forme linéaire (3.8) est à minimiser, Λ étant fixé et vérifiant les restrictions (3.10), le nombre λ doit être choisi le plus petit possible.

Par conséquent le système de conditions (3.9) peut être remplacé par la relation

$$\max_{1 \leqslant \nu \leqslant N_1} \{\sigma_\nu - (P_\nu, \Lambda)\} = \lambda. \tag{3.11}$$

D'après les définitions du nombre σ_ν et du vecteur P_ν

$$\sigma_\nu - (P_\nu, \Lambda) = (C, X_\nu) - (A^{(0)}X_\nu, \Lambda) =$$
$$= (C - \Lambda A^{(0)}, X_\nu) = (C_\Lambda, X_\nu),$$

où, de même qu'auparavant [cf. (1.25)], le vecteur

$$C_\Lambda = C - \Lambda A^{(0)}.$$

La relation (3.11) peut donc être mise sous la forme

$$\max_{1 \leqslant \nu \leqslant N_1} (C_\Lambda, X_\nu) = \lambda. \tag{3.12}$$

Ces raisonnements montrent que le problème de minimisation de la fonction

$$\max_{1 \leqslant \nu \leqslant N_1} (C_\Lambda, X_\nu) + (B^{(0)}, \Lambda) \tag{3.13}$$

sous les conditions (3.10) est équivalent au $\widetilde{Z}$-problème (3.8)-(3.10).

3.2. Soit R l'ensemble des points Λ tels que

$$\sup_{X \in G} (C_\Lambda, X) = \max_{X \in G} (C_\Lambda, X) < \infty. \tag{3.14}$$

Introduisons dans la discussion la fonction $f(\Lambda)$ définie sur l'ensemble R par la relation

$$f(\Lambda) = \max_{X \in G} (C_\Lambda, X) + (B^{(0)}, \Lambda). \tag{3.15}$$

Vérifions que R, domaine de définition de la fonction $f(\Lambda)$, est un ensemble convexe auquel correspond un système de conditions du type (3.10). A cette fin, récrivons les inégalités (3.10) sous une forme équivalente

$$(C_\Lambda, X_\nu) \leqslant 0, \quad \nu = N_1 + 1, \ldots, N. \tag{3.16}$$

Tout point $X \in G$ peut être représenté sous la forme

$$X = \sum_{\nu=1}^{N} X_\nu z_\nu, \quad \sum_{\nu=1}^{N_1} z_\nu = 1, \quad z_\nu \geqslant 0,$$
$$\nu = 1, 2, \ldots, N. \tag{3.17}$$

Par suite, si les conditions (3.16) sont respectées,

$$(C_\Lambda, X) \leqslant \max_{1 \leqslant \nu \leqslant N_1} (C_\Lambda, X_\nu) < \infty.$$

Mais si pour un certain numéro ν_0 situé entre $N + 1$ et N

$$(C_\Lambda, X_{\nu_0}) > 0,$$

alors, en posant $X(\theta) = X_1 + \theta X_{\nu_0}$, il vient

$$\lim_{\theta \to \infty} (C_\Lambda, X(\theta)) = \infty.$$

Etant donné que $X(\theta) \in G$ pour tout $\theta \geqslant 0$,

$$\sup_{X \in G} (C_\Lambda, X) = \infty.$$

Ainsi, l'ensemble R est en effet défini par le système d'inégalités (3.10).

Si G est un polyèdre convexe, alors $N_1 = N$, c'est-à-dire le système de conditions (3.10) n'existe pas. Dans ce cas, R se confond avec l'espace à m dimensions des points Λ tout entier. Il se peut que R ne contienne aucun point. Ce cas est équivalent à l'incompatibilité des conditions du $\widetilde{Z}$-problème. Ci-dessous, on suppose que R est un ensemble non vide.

Soit $\Lambda \in R$. La présentation (3.17) entraîne

$$f(\Lambda) - (B^{(0)}, \Lambda) = \max_{X \in G} (C_\Lambda, X) = \max_{1 \leqslant \nu \leqslant N_1} (C_\Lambda, X_\nu). \qquad (3.18)$$

C'est pourquoi le problème (3.13), (3.10), équivalent au $\widetilde{Z}$-problème (3.8)-(3.10), se ramène à la minimisation de la fonction $f(\Lambda)$ définie sur l'ensemble convexe R.

Si

$$\inf_{\Lambda \in R} f(\Lambda) = -\infty,$$

alors il résulte des éléments de la théorie de la dualité que le Z-problème (3.5)-(3.7) n'a aucun programme, c'est-à-dire que les conditions (3.2)-(3.4) du X-problème initial sont contradictoires. Mais si

$$\inf_{\Lambda \in R} f(\Lambda) = \min_{\Lambda \in R} f(\Lambda) = f(\Lambda^*) > -\infty,$$

d'après (3.12) et (3.18) le vecteur

$$\overline{\Lambda}^* = (\Lambda^*, \lambda^*)$$

avec

$$\lambda^* = f(\Lambda^*) - (B^{(0)}, \Lambda^*)$$

est solution du $\widetilde{Z}$-problème (3.8)-(3.10).

Dans l'espace de dimension $(m+1)$ des points (Λ, λ_{m+1}), chaque sommet X_ν $(\nu = 1, 2, \ldots, N_1)$ de l'ensemble convexe G engendre un hyperplan H_ν défini par l'équation

$$\lambda_{m+1} + (A^{(0)}X_\nu - B^{(0)}, \Lambda) = (C, X_\nu). \qquad (3.19)$$

Cette équation peut s'écrire sous une forme équivalente

$$\lambda_{m+1} = (C_\Lambda, X_\nu) + (B^{(0)}, \Lambda). \qquad (3.20)$$

Dans ce même espace, le vecteur directeur X_ν $(\nu = N_1 + 1, \ldots, N)$ de chaque arête non bornée de l'ensemble convexe G engendre un hyperplan H_ν défini par l'équation

$$(A^{(0)}X_\nu, \Lambda) = (C, X_\nu) \qquad (3.21)$$

ou, ce qui revient au même,

$$(C_\Lambda, X_\nu) = 0. \tag{3.22}$$

L'hyperplan H_ν coupe l'axe des λ_{m+1} avec $1 \leqslant \nu \leqslant N_1$, il est parallèle à cet axe avec $N_1 + 1 \leqslant \nu \leqslant N$.

Déterminons dans l'espace considéré l'ensemble $\overline{R}$ composé des points (Λ, λ_{m+1}) tels que

$$\lambda_{m+1} \geqslant f(\Lambda), \quad \Lambda \in R. \tag{3.23}$$

En vertu de (3.18) la première condition du système (3.23) peut s'écrire

$$\lambda_{m+1} \geqslant (C_\Lambda, X_\nu) + (B^{(0)}, \Lambda), \quad \nu = 1, 2, \ldots, N_1. \tag{3.24}$$

On sait que la deuxième relation (3.23) est équivalente au système de conditions (3.10) ou, ce qui revient au même, à (3.16). Par conséquent, $\overline{R}$ est un ensemble convexe qui coïncide avec la partie commune des demi-espaces (3.24) et (3.16) engendrés par les hyperplans H_ν ($\nu = 1, 2, \ldots, N$). Il résulte des conditions (3.23), qui définissent l'ensemble $\overline{R}$, que le problème de minimisation de la fonction $f(\Lambda)$ sur l'ensemble R est équivalent au choix du point « le plus bas » (au sens de l'axe des λ_{m+1}) de l'ensemble $\overline{R}$.

La surface

$$\lambda_{m+1} = f(\Lambda)$$

coïncide avec une partie de la frontière de l'ensemble convexe $\overline{R}$. Par conséquent $f(\Lambda)$ est une fonction linéaire par morceaux convexe vers le bas.

Résumons les conclusions ainsi obtenues.

Théorème 3.1. *L'exploration du $\widetilde{Z}$-problème* (3.8)-(3.10) *est équivalente soit*

1) *à la minimisation de la fonction linéaire par morceaux convexe vers le bas $f(\Lambda)$ donnée sur l'ensemble convexe $\overline{R}$, où $f(\Lambda)$ et R sont définis par les conditions* (3.15) *et* (3.16) *respectivement; soit*

2) *au choix du point « le plus bas » (au sens de l'axe des λ_{m+1}) de l'ensemble convexe $\overline{R}$, partie commune des demi-espaces* (3.24) *et* (3.16).

Si R se confond avec l'espace de dimension m tout entier, c'est-à-dire si G est un polyèdre convexe, il est plus commode d'utiliser la première interprétation du Z-problème. Dans le cas où R est une partie de l'espace de dimension m des points Λ, c'est-à-dire où G est un ensemble non borné, il est préférable de faire appel à deuxila ème interprétation.

Le problème de la minimisation de la fonction $f(\Lambda)$ sur l'ensemble R ou, ce qui revient au même, le problème de la recherche du point le plus bas de l'ensemble $\overline{R}$ est la formulation duale du problème (3.1)-(3.4) décomposé en blocs.

3.3. Lors de l'analyse du $\widetilde{Z}$-problème, il est naturel de considérer seulement les points de l'ensemble $\overline{R}$ situés à la surface

$$\lambda_{m+1} = f(\Lambda). \tag{3.25}$$

Soit $\overline{\Lambda}' = (\Lambda', \lambda'_{m+1}) \in \overline{R}$ un point quelconque de la surface (3.25). Prélevons parmi les hyperplans H_ν ceux qui passent par le point $\overline{\Lambda}'$. Il résulte des équations (3.20), (3.22) et de la définition de la fonction $f(\Lambda)$ que le vecteur X_ν qui engendre un des hyperplans cherchés doit vérifier la condition

$$\lambda'_{m+1} = (C_{\Lambda'}, X_\nu) + (B^{(0)}, \Lambda') = \max_{X \in G} (C_{\Lambda'}, X) + (B^{(0)}, \Lambda') \tag{3.26}$$

$$\text{si } 1 \leqslant \nu \leqslant N_1$$

et

$$(C_{\Lambda'}, X_\nu) = 0 \tag{3.27}$$

$$\text{si } N_1 + 1 \leqslant \nu \leqslant N.$$

Soit G_Λ l'ensemble des solutions du X_Λ-problème

$$(C_\Lambda, X) \to \max, \tag{3.28}$$

$$X \in G. \tag{3.29}$$

Le point X_ν qui vérifie l'égalité (3.26) est un sommet de l'ensemble convexe $G_{\Lambda'}$.

Le vecteur X_ν, qui obéit à la condition (3.27), est parallèle à l'une des arêtes non bornées de l'ensemble $G_{\Lambda'}$.

Ainsi les hyperplans H_ν qui contiennent le point $\overline{\Lambda}'$ situé à la surface (3.25) sont engendrés par les sommets et les vecteurs directeurs des arêtes non bornées de l'ensemble $G_{\Lambda'}$ des solutions du $X_{\Lambda'}$-problème. Il est évident que la réciproque est vraie également : l'hyperplan H_ν engendré par l'un des éléments de l'ensemble $G_{\Lambda'}$ contient le point $\overline{\Lambda}'$.

Soit H un hyperplan arbitraire, un hyperplan d'appui pour l'ensemble $\overline{R}$ en $\overline{\Lambda}'$ et non parallèle à l'axe des λ'_{m+1}. Son équation s'écrit

$$\lambda_{m+1} + (S, \Lambda - \Lambda') = \lambda'_{m+1}, \tag{3.30}$$

où $S = (s_1, s_2, \ldots, s_m)$.

H étant un hyperplan d'appui, il en résulte que son vecteur directeur $\overline{S} = (S, 1)$ est représenté sous la forme d'une combinaison non négative des vecteurs directeurs des hyperplans H_ν passant par le point $\overline{\Lambda}'$. Le vecteur directeur de l'hyperplan H_ν défini par l'équation (3.19) est égal à

$$\overline{S}_\nu = (A^{(0)} X_\nu - B^{(0)}, 1).$$

Le vecteur directeur de l'hyperplan H_ν vérifiant l'équation (3.21) est

$$\overline{S}_\nu = (A^{(0)}X_\nu,\ 0).$$

Par suite,

$$S = \sum_{\nu=1}^{N_1} z_\nu (A^{(0)}X_\nu - B^{(0)}) + \sum_{\nu=N_1+1}^{N} z_\nu A^{(0)}X_\nu, \tag{3.31}$$

$$1 = \sum_{\nu=1}^{N_1} z_\nu,\ z_\nu \geqslant 0, \quad \nu = 1,\ 2,\ \ldots,\ N. \tag{3.32}$$

L'égalité (3.31) peut être mise sous la forme

$$S = A^{(0)}X - B^{(0)},$$

où

$$X = \sum_{\nu=1} z_\nu X_\nu. \tag{3.33}$$

Constatons que si l'hyperplan H_ν ne passe pas par le point $\overline{\Lambda}'$ dans la relation (3.31) $z_\nu = 0$. C'est pourquoi le point X défini par les égalités (3.33) et (3.32) appartient à l'ensemble $G_{\Lambda'}$.

D'autre part, on vérifie aisément que l'hyperplan

$$\lambda_{m+1} + (A^{(0)}X - B^{(0)},\ \Lambda - \Lambda') = \lambda'_{m+1} \tag{3.34}$$

défini par un point quelconque $X \in G_{\Lambda'}$ est un hyperplan d'appui pour l'ensemble $\overline{R}$ en $\overline{\Lambda}'$ (cf. exercice 7).

Réunissons en un ensemble $H(\overline{\Lambda})$ les hyperplans d'appui pour $\overline{R}$ en $\overline{\Lambda}$ et qui coupent l'axe des λ_{m+1}. Les conclusions obtenues peuvent être énoncées sous la forme de la proposition suivante.

Théorème 3.2. *L'ensemble $H(\overline{\Lambda}')$ se compose d'hyperplans du type* (3.34), *où $X \in G_{\Lambda'}$.*

A partir du théorème 3.2 on obtient sans peine l'expression de l'accroissement de la fonction $f(\Lambda)$ avec une variation suffisamment petite de son argument.

Supposons que les points Λ' et $\Lambda' + l$ appartiennent à R. Dans un voisinage suffisamment petit du point Λ' l'hypersurface (3.25) se compose de morceaux d'hyperplans appartenant à l'ensemble $H(\overline{\Lambda}')$.

C'est pourquoi, pour un $\varepsilon_0 > 0$ suffisamment petit, le théorème 3.2 conduit à la relation

$$f(\Lambda' + \varepsilon l) = \max_{X \in G_{\Lambda'}} [\lambda'_{m+1} - (A^{(0)}X - B^{(0)},\ \varepsilon l)],$$

où

$$0 \leqslant \varepsilon \leqslant \varepsilon_0.$$

Cette dernière relation peut s'écrire

$$\frac{f(\Lambda'+\varepsilon l)-f(\Lambda')}{\varepsilon} = \max_{X\in G_{\Lambda'}} (B^{(0)}-A^{(0)}X,\ l). \tag{3.35}$$

L'utilisation de la formule (3.35) n'implique pas l'obligation de supposer d'avance que $\Lambda' + l \in R$. Si le deuxième membre de (3.35) est fini, avec $\varepsilon > 0$ suffisamment petit le point $\Lambda' + \varepsilon l \in R$ et la formule (3.35) a lieu. Mais si le deuxième membre de (3.35) est infini, alors, quel que soit $\varepsilon > 0$, le point $\Lambda' + \varepsilon l \notin R$. Nous laissons au lecteur le soin de le démontrer (cf. exercice 8).

3.4. Il est évident que le point $\bar{\Lambda}^* = (\Lambda^*, \lambda^*_{m+1})$ est le point « le plus bas » cherché de l'ensemble convexe $\bar{R}$ si et seulement si l'hyperplan

$$\lambda_{m+1} = \lambda^*_{m+1}$$

appartient à l'ensemble $H(\bar{\Lambda}^*)$, c'est-à-dire s'il est d'appui à R en $\bar{\Lambda}^*$. D'après le théorème 3.2, ceci équivaut à l'existence d'un vecteur X^* vérifiant les conditions

$$X^* \in G_{\Lambda^*}, \tag{3.36}$$

$$A^{(0)}X^* = B^{(0)}. \tag{3.37}$$

On voit aisément que le vecteur X^*, justifié par les restrictions (3.36) et (3.37), est la solution du problème (3.1)-(3.4). En effet, puisque X^* est la solution du X_{Λ^*}-problème vérifiant la condition (3.37), on a pour tout programme X du problème (3.1)-(3.4)

$$(C,\ X) \leqslant f(\Lambda^*) = (C_{\Lambda^*},\ X^*) + (B^{(0)},\ \Lambda^*) =$$
$$= (C,\ X^*) + (B^{(0)} - A^{(0)}X^*,\ \Lambda^*) = (C,\ X^*).$$

Dans la méthode de décomposition de Dantzig-Wolfe, le déplacement vers l'optimum s'accomplit suivant les couples de points $(\Lambda,\ X)$ qui vérifient la condition (3.37). La relation (3.36) est testée au moyen de la résolution du X_Λ-problème. Dès qu'elle se trouve vérifiée, le programme optimal cherché est obtenu.

Voici l'interprétation géométrique de la méthode de décomposition. Soit un programme de base du Z-problème, à base composée des vecteurs

$$\bar{P}_{i_s} + \begin{pmatrix} A^{(0)}X_{i_s} \\ \varepsilon_{i_s} \end{pmatrix}, \quad s = 1,\ 2,\ \ldots,\ m+1,$$

et aux composantes de base z_{s0}.

A chaque point X_{i_s} correspond un hyperplan H_{i_s} du type (3.19) si X_{i_s} est un sommet de G, et du type (3.21) si X_{i_s} est le vecteur directeur d'une arête infinie de G.

Soit $(\Lambda, \lambda^{(1)})$ le vecteur des critères des conditions du Z-problème par rapport à la base donnée.

Posons

$$\lambda = \lambda^{(1)} + (B^{(0)}, \Lambda).$$

On vérifie facilement que $\overline{\Lambda} = (\Lambda, \lambda)$ est un point commun aux hyperplans H_{i_s} (cf. exercice 9).

Si le point $\overline{\Lambda} \in \overline{R}$, c'est-à-dire si

$$\Lambda \in R, \quad \lambda = f(\Lambda), \tag{3.38}$$

$X = \sum_{s=1}^{m+1} z_{s0} X_{i_s}$ est un programme optimal du problème (3.1)-(3.4) et $\overline{\Lambda}$, le point « le plus bas » de l'ensemble $\overline{R}$.

Les conditions (3.38) se vérifient au cours de la discussion du X_Λ-problème.

Si la solution du X_Λ-problème est irréalisable, la première condition du système (3.38) est violée. Lorsqu'on l'établit, il faut ajouter à la base du Z-problème le vecteur associé à l'un des hyperplans du type (3.21), qui séparent le point $\overline{\Lambda}$ de l'ensemble $\overline{R}$.

Si le X_Λ-problème est possible, mais la valeur calculée

$$f(\Lambda) > \lambda,$$

il faut introduire dans la base du Z-problème le vecteur correspondant à un certain hyperplan du type (3.19) qui sépare $\overline{\Lambda}$ de $\overline{R}$.

Ainsi la procédure de résolution du problème (3.1)-(3.4) par la méthode de décomposition consiste à se déplacer vers le point « le plus bas » de l'ensemble $\overline{R}$ suivant les points $\overline{\Lambda} = (\Lambda, \lambda) \notin \overline{R}$. Par ailleurs, dans le cas où le problème (3.1)-(3.4) est réalisable, tout point intermédiare $\overline{\Lambda}$ se situe « en-dessous » de l'hyperplan

$$\lambda_{m+1} = f(\Lambda^*),$$

la distance à cet hyperplan diminuant de façon monotone. ($\overline{\Lambda}^* = (\Lambda^*, f(\Lambda^*))$ est ici le point « le plus bas » de l'ensemble $\overline{R}$.)

3.5. La formulation duale du problème décomposé en blocs conduit à l'approche suivante, qui est une façon naturelle de résoudre le problème (3.1)-(3.4). D'après cette formulation, la discussion du problème initial se ramène à la minimisation de la fonction linéaire par morceaux convexe vers le bas $f(\Lambda)$, définie sur un certain ensemble convexe R.

Voici les traits particuliers du problème de minimisation de la fonction $f(\Lambda)$;

premièrement, $f(\Lambda)$ est donnée seulement par l'algorithme (pour calculer sa valeur en Λ, il faut résoudre le X_Λ-problème),

et deuxièmement, cette fonction n'est pas dérivable en tous les points et par conséquent, dans le cas considéré, les méthodes de minimisation des fonctions lisses ne sont pas applicables.

Considérons le schéma général de la procédure de minimisation de la fonction $f(\Lambda)$. Choisissons un certain point de départ $\Lambda' \in R$ et cherchons une direction (un vecteur) l telle que, premièrement, $\Lambda' + \varepsilon l \in R$ avec $\varepsilon > 0$ suffisamment petit et, deuxièmement, la valeur de la fonction $f(\Lambda)$ décroisse le long de l. Nous dirons qu'une telle direction est *possible*. D'après la formule (3.35), une condition nécessaire et suffisante pour que la direction l observée soit possible consiste à vérifier l'inégalité

$$\max_{X \in G_{\Lambda'}} (B^{(0)} - A^{(0)}X,\ l) < 0. \tag{3.39}$$

Désignons par $L_{\Lambda'}$ l'ensemble des directions qui justifient la condition (3.39). Supposons que nous ayons à notre disposition quelque procédé T de choix de $l \in L_{\Lambda'}$ pour n'importe quel ensemble $L_{\Lambda'}$ non vide. Sélectionnons à l'aide du procédé T la direction $l \in L_{\Lambda'}$ ou constatons que l'ensemble $L_{\Lambda'}$ des directions possibles est vide. Dans ce dernier cas, Λ' est le point cherché du minimum de la fonction $f(\Lambda)$. Mais si la direction $l' \in L_{\Lambda'}$ est trouvée, déplaçons-nous au point

$$\Lambda(l,\ \theta) = \Lambda' + \theta l'. \tag{3.40}$$

Soumettons le choix du nombre $\theta > 0$ participant à la formation du point (3.40) à une restriction naturelle: avec $0 \leqslant t < \theta$, la fonction de l'argument scalaire t

$$\varphi(t) = f(\Lambda' + tl)$$

doit être décroissante monotone.

Désignons par E le mode de choix du nombre θ qui satisfait à cette dernière condition. Le point $\Lambda'' = \Lambda' + \theta l$ étant obtenu, nous lui appliquerons les mêmes raisonnements, etc. Ainsi, si l'on se donne certains procédés T de choix d'une direction possible et E de définition du nombre θ, on obtient une suite décroissante monotone $\{f(\Lambda^{(k)})\}$.

Nous donnerons dans les paragraphes qui suivent une série de procédés T et E qui aboutissent à la borne inférieure cherchée de la fonction $f(\Lambda)$, c'est-à-dire tels que

$$\lim_{k \to \infty} f(\Lambda^{(k)}) = \inf_{\Lambda \in R} f(\Lambda).$$

Nous allons voir que chaque couple de tels procédés définit une certaine méthode de programmation par blocs.

Sur la base de ces procédés on peut également former des méthodes combinées de programmation par blocs qui mettent à profit, pour résoudre le problème, les différents procédés T et E.

Le trait commun aux méthodes de programmation par blocs qui cadrent avec le schéma établi est que, contrairement à la méthode de décomposition, leur réalisation impose un déplacement suivant une surface (3.25).

§ 4. Analogue dual de la méthode de décomposition

4.1. La première procédure de résolution du problème décomposé en blocs (3.1)-(3.4) que nous allons examiner est associé à un déplacement suivant les sommets d'une surface polyédrique (3.25) ou, ce qui revient au même, suivant les sommets d'un ensemble convexe $\overline{R}$ soumis aux conditions (3.10), (3.24). Pour commencer la procédure, il faut connaître $m+1$ hyperplans H_{s_i} $(i = 1, 2, \ldots, m+1)$ dont l'intersection détermine le sommet recherché $\overline{\Lambda}' = (\Lambda', \lambda')$.

La méthode d'obtention de ce système d'hyperplans est décrite au § 6. Le vecteur directeur n_ν de l'hyperplan H_ν, orienté vers l'intérieur de l'ensemble $\overline{R}$, est d'après les relations (3.10) et (3.24) de la forme

$$n_\nu = \begin{pmatrix} A^{(0)}X_\nu - \varepsilon_\nu B^{(0)} \\ \varepsilon_\nu \end{pmatrix}, \tag{4.1}$$

où

$$\varepsilon_\nu = \begin{cases} 1 & \text{si} \quad \nu \leqslant N_1, \\ 0 & \text{si} \quad \nu > N_1. \end{cases} \tag{4.2}$$

Soit $\overline{l}_i = (l_{i1}, l_{i2}, \ldots, l_{i,m+1})$ le vecteur orienté vers l'intérieur de l'ensemble $\overline{R}$ parallèlement à l'intersection des hyperplans

$$H_{s_1}, \ldots, H_{s_{i-1}}, H_{s_{i+1}}, \ldots, H_{s_{m+1}}.$$

Il est clair qu'après une normalisation correspondante

$$(n_{s_\alpha}, \overline{l}_i) = \begin{cases} 0 & \text{si} \quad \alpha \neq i, \\ 1 & \text{si} \quad \alpha = i. \end{cases} \tag{4.3}$$

Par conséquent, la matrice

$$\| l_{ij} \|_{m+1} = (n_{s_1}, n_{s_2}, \ldots, n_{s_{m+1}})^{-1}. \tag{4.4}$$

Si la dernière composante $l_{i,m+1}$ de la direction $\overline{l}_i = (l_i, l_{i,m+1})$ est non négative, le déplacement du point Λ' dans la direction l_i ne diminue pas la valeur de la fonction $f(\Lambda)$. Si cette condition est vérifiée pour tout i, c'est-à-dire si

$$l_{i,m+1} \geqslant 0 \quad \text{pour} \quad i = 1, 2, \ldots, m+1, \tag{4.5}$$

le point Λ' est le minimum recherché. Cette conclusion résulte du critère connu d'optimalité du programme utilisé dans la méthode primale du simplexe.

Supposons que l'ensemble convexe $\overline{R}$ soit non dégénéré, c'est-à-dire qu'en chacun de ses sommets se coupent exactement $m+1$ hyperplans H_ν. Dans ce cas, pour tout $1 \leqslant i \leqslant m+1$, la demi-droite $\overline{\Lambda}' + \theta \overline{l}_i$, $\theta \geqslant 0$, contient une arête de l'ensemble $\overline{R}$, issue du point $\overline{\Lambda}'$. C'est pourquoi il existe un $\theta_0 > 0$ tel que

$$\overline{\Lambda}' + \theta \overline{l}_i \in \overline{R} \quad \text{si} \quad 0 \leqslant \theta \leqslant \theta_0. \tag{4.6}$$

Supposons que la condition (4.5) ne soit pas respectée, c'est-à-dire que pour certains i

$$l_{i,m+1} < 0. \tag{4.7}$$

En vertu de (4.6), n'importe laquelle de ces directions est possible, puisque avec $\theta > 0$ suffisamment petit

$$\lambda'_{m+1} = f(\Lambda') > f(\Lambda' + \theta l_i) = \lambda'_{m+1} + \theta l_{i,m+1}.$$

Le calcul des directions l_i et le choix de celles d'entre elles qui sont possibles peuvent se faire non seulement d'après les relations (4.4) mais encore à l'aide de la matrice

$$\| e_{ij} \|_{m+1} = (\overline{P}_{s_1}, \overline{P}_{s_2}, \ldots, \overline{P}_{s_{m+1}})^{-1},$$

où $\overline{P}_\nu = \begin{pmatrix} P_\nu \\ \varepsilon_\nu \end{pmatrix}$ est le vecteur des conditions du Z-problème (3.5)-(3.7) correspondant au vecteur X_ν (au sommet de G ou au vecteur directeur d'une arête non bornée de cet ensemble).

Nous laissons au lecteur le soin de vérifier (cf. exercice 10) que

$$l_{ij} = \begin{cases} e_{ij} & \text{si } j \leqslant m, \\ e_{i0} & \text{si } j = m+1, \end{cases} \tag{4.8}$$

où les e_{i0} sont les coefficients de décomposition du vecteur des contraintes $\overline{B}^{(0)} = \begin{pmatrix} B_0 \\ 1 \end{pmatrix}$ du problème (3.5)-(3.7) par rapport au système de vecteurs $\overline{P}_{s_i}$, $i = 1, 2, \ldots, m+1$.

4.2. Choisissons une des directions qui obéissent à la restriction (4.7) (par exemple, des conditions minimisant la dernière composante). Supposons que cette direction soit $\overline{l}_r$. La grandeur θ' de l'étape qu'il convient de faire le long de la direction choisie $\overline{l}_r$ est définie par la condition de passage de l'ensemble $\overline{R}$ au sommet Λ'', voisin de $\overline{\Lambda}'$. Pour choisir θ', considérons le problème paramétrique qui consiste à maximiser la forme linéaire

$$(C_{\Lambda'} - \theta l_r A^{(0)}, X), \tag{4.9}$$

qui dépend du paramètre $\theta \geqslant 0$, sous la condition

$$X \in G. \tag{4.10}$$

Ayant choisi comme programme de base initial l'un des sommets X_{s_i} de l'ensemble G, qui engendrent les hyperplans H_{s_i} $(i \neq r)$, appliquons au problème (4.9), (4.10) la méthode de programmation paramétrique exposée au § 1 du chapitre 3.

On sait qu'une étape distincte de cette méthode coïncide avec une itération de la méthode primale du simplexe, lorsque le choix du vecteur à inclure dans la base se fait en se guidant par une règle spéciale. La procédure de résolution du problème (4.9), (4.10) se poursuit jusqu'à mise en évidence de l'une des éventualités suivantes.

1°. On a trouvé que θ_1 est un point critique positif minimal et on a construit le programme de base X_k du problème (4.9), (4.10), optimal avec $\theta_1 \leqslant \theta \leqslant \theta_2$, $\theta_2 > \theta_1$.

2°. On a établi que pour $0 \leqslant \theta \leqslant \theta_1$ le problème (4.9), (4.10) a une solution réalisable et, de plus, que $[0, \theta_1]$ ne compte pas de points critiques, alors que pour $\theta > \theta_1$ la forme linéaire (4.9) est non bornée sur l'ensemble G. On a construit simultanément le vecteur directeur X_k de l'arête non bornée de G sur laquelle (4.9) est non bornée avec $\theta > \theta_1$.

3°. Le problème (4.9), (4.10) est réalisable sur la demi-droite $\theta \geqslant 0$ qui ne contient pas de valeurs critiques. Posons dans ce cas $\theta_1 = \infty$.

Il résulte de l'hypothèse de non-dégénérescence de l'ensemble convexe $\overline{R}$ que

$$f(\Lambda' + \theta l_r) = f(\Lambda') + \theta l_{r,m+1}$$

pour tout θ fini vérifiant la condition

$$0 \leqslant \theta \leqslant \theta_1. \tag{4.11}$$

Par conséquent, le cas 3° pour lequel $\theta_1 = \infty$ conduit à la relation

$$\inf_{\Lambda \in R} f(\Lambda) = -\infty$$

qui traduit l'incompatibilité des conditions du problème (3.1)-(3.4) (cf. paragraphe 3.2).

Supposons que la discussion du problème (4.9), (4.10) se soit achevée par le cas 1° ou 2°. L'interprétation géométrique de ce fait est donnée par le déplacement le long de l'arête de l'ensemble $\overline{R}$ au vecteur directeur $\overline{l}_r$ qui conduit à l'hyperplan H_k engendré par le vecteur X_k. Dans le cas 1°, le vecteur X_k correspond à un sommet de l'ensemble G; dans le cas 2°, le vecteur X_k est dirigé le long de l'arête non bornée de l'ensemble G. Posons

$$\overline{\Lambda}'' = \overline{\Lambda}' + \theta' \overline{l}_r, \quad \theta' = \theta_1.$$

Le nouveau sommet $\overline{\Lambda}''$ est formé par l'intersection des hyperplans

$$H_{s_1}, H_{s_2}, \ldots, H_{s_{r-1}}, H_k, H_{s_{r+1}}, \ldots, H_{s_{m+1}}.$$

L'étape suivante s'effectue selon les mêmes règles, en partant du point $\overline{\Lambda}''$. Avant de commencer l'étape, la matrice (4.4) est transformée en une nouvelle matrice, dans laquelle le vecteur n_{s_r} est remplacé par

$$n_k = \begin{pmatrix} A^{(0)}X_k - \varepsilon_k B^{(0)} \\ \varepsilon_k \end{pmatrix}.$$

Pour ce faire, on recourt aux formules de récurrence usuelles de la programmation linéaire.

4.3. Nous avons déjà dit que dans le cas non dégénéré $\theta' > 0$, et à chaque étape de la méthode la valeur de la fonction $f(\Lambda)$ diminue. En cas de dégénérescence, θ' peut être nul. Alors, $\overline{\Lambda}'' = \overline{\Lambda}'$. Le résultat de cette étape consiste seulement à passer à un nouveau système de $m+1$ hyperplans qui se coupent en $\overline{\Lambda}'$. Pour rendre possible la réalisation d'une étape de cette procédure, sans supposer la non-dégénérescence de l'ensemble $\overline{R}$, il faut compléter les conditions des cas 1°, 2°, 3°.

Par définition, θ_1 est un nombre non négatif qui ne peut devenir nul que dans le cas 2°. Lorsque $\theta_1 > 0$, nous désignerons par $X_{k'}$ la solution de base du problème (4.9), (4.10) pour toutes les valeurs finies de θ comprises entre zéro et θ_1.

Posons

$$\alpha = -(l_r A^{(0)}, X_{k'}).$$

Il est clair que $\alpha \geqslant l_{r,m+1}$.

Pour $\alpha = l_{r,m+1}$ (ce qui est toujours vrai pour une situation non dégénérée) ou pour $\theta_1 = 0$, la procédure est définie par les règles décrites dans ce qui précède.

Mais si

$$\theta_1 > 0 \quad \text{et} \quad \alpha > l_{r,m+1},$$

l'hyperplan engendré par le vecteur $X_{k'}$ vient remplacer l'hyperplan H_{s_r}. Il est évident que dans ce dernier cas $\theta' = 0$, c'est-à-dire $\overline{\Lambda}' = \overline{\Lambda}''$.

La règle de sélection du vecteur X_k ou $X_{k'}$ qui dans la méthode considérée rend tout à fait impossible le cyclage est plutôt trop complexe; de plus, comme toutes les règles analogues, elle n'a pas une grande valeur pratique. C'est pourquoi nous n'en donnons pas la description.

4.4. En utilisant la formule (4.8), il est facile de vérifier que la méthode de programmation par blocs décrite ici se ramène à la résolution du Z-problème par la méthode duale du simplexe, c'est-à-dire constitue l'analogue dual de la méthode de décomposition. La matrice $(\overline{P}_{s_1}, \ldots, \overline{P}_{s_{m+1}})$ est la base d'un pseudoprogramme du Z-problème. La composante négative de décomposition du vecteur

$\overline{B}^{(0)}$ par rapport à la base du pseudoprogramme détermine le vecteur $\overline{P}_{s_r}$ à éliminer de la base. Le vecteur des conditions du Z-problème qui remplace $\overline{P}_{s_r}$ dans la base est recherché au cours de la discussion du problème (4.9), (4.10).

Une méthode apparentée à celle décrite dans ce paragraphe a été proposée à partir d'autres considérations dans [1].

La différence entre ces deux méthodes ne se manifeste qu'à l'étape de définition de la quantité θ'. Dans [1], on étudie à cet effet un problème de programmation homographique. Le mode de définition de θ' proposé dans [1] se ramène à l'analyse du problème paramétrique (4.9), (4.10) en partant de la valeur initiale $\theta = \infty$. Ce mode de recherche de θ' étant lié au calcul de toutes les valeurs critiques positives, la méthode examinée ici et dans laquelle la valeur initiale $\theta = 0$ doit être reconnue comme plus efficace.

Constatons que la méthode citée constitue l'une des applications du schéma général exposé au paragraphe 3.5 avec des méthodes particulières T et E.

§ 5. Autres méthodes de programmation par blocs fondées sur la minimisation de la fonction $f(\Lambda)$

5.1. Nous avons déjà dit au paragraphe 3.5 que toute méthode de programmation par blocs liée à la minimisation de la fonction $f(\Lambda)$ repose sur un ou plusieurs modes définis du choix de la direction possible. Dans l'analogue dual de la méthode de décomposition exposée au § 4 ce choix se fait parmi les arêtes de l'ensemble convexe $\overline{R}$.

Nous donnons ci-dessous un schéma général de formation de procédés T qui font partie d'une certaine classe assez large. Soit l une direction arbitraire issue du point Λ'. Introduisons la fonction

$$\varphi_1(l) = \sup_{X \in G_{\Lambda'}} (B^{(0)} - A^{(0)}X,\ l). \tag{5.1}$$

Ainsi qu'il résulte de la relation (3.39), le choix d'une direction possible est équivalent à la construction d'un vecteur l satisfaisant à l'inégalité

$$\varphi_1(l) < 0. \tag{5.2}$$

Choisissons un ensemble convexe arbitraire M de points $l = (l_1, l_2, \ldots, l_m)$ vérifié par deux restrictions:

1) le point $0 = (0, 0, \ldots, 0)$ est un point intérieur de M; (5.3)

2) $v_0 = \inf_{l \in M} \varphi_1(l) > -\infty$. (5.4)

L'ensemble M et le point Λ' engendrent le problème $\Gamma(M, \Lambda')$ qui consiste à minimiser la fonction $\varphi_1(l)$ sous la condition $l \in M$. Le problème $\Gamma(M, \Lambda)$ se ramène facilement à un problème de programmation linéaire. En effet, soient $X_1, X_2, \ldots, X_{\gamma_1}$ les sommets de l'ensemble $G_{\Lambda'}$, $X_{\gamma_1+1}, X_{\gamma_1+2}, \ldots, X_\gamma$ les vecteurs directeurs des arêtes non bornées de $G_{\Lambda'}$. Supposons également que le système de conditions qui définit M s'écrit

$$\sum_{j=1}^{m} d_{ij} l_j \leqslant d_i, \quad i=1, 2, \ldots, r. \tag{5.5}$$

Il est aisé de voir (cf. exercice 11) que le problème $\Gamma(M, \Lambda')$ est équivalent au problème de programmation linéaire qui consiste à minimiser

$$v \tag{5.6}$$

sous les conditions

$$(B^{(0)} - A^{(0)} X_i, l) \leqslant v, \quad i=1, 2, \ldots, \gamma_i; \tag{5.7}$$

$$-(A^{(0)} X_i, l) \leqslant 0, \quad i=\gamma_1+1, \ldots, \gamma; \tag{5.8}$$

$$\sum_{j=1}^{m} d_{ij} l_j \leqslant d_i, \quad i=1, 2, \ldots, r. \tag{5.9}$$

La condition (5.3) assure la compatibilité des contraintes (5.7)-(5.9). La restriction (5.4) impose à la forme linéaire (5.6) d'être bornée inférieurement sur l'ensemble (5.7)-(5.9). Par conséquent, le problème (5.6)-(5.9) a une solution réalisable, de même que le problème $\Gamma(M, \Lambda')$, soit

$$v_0 = \inf_{l \in M} \varphi_1(l) = \varphi_1(l^*),$$

et de plus, en vertu de (5.3),

$$v_0 \leqslant 0. \tag{5.10}$$

Introduisons ensuite dans la discussion le problème $\widetilde{\Gamma}(M, \Lambda')$, dual au $\Gamma(M, \Lambda')$. L'application au $\Gamma(M, \Lambda')$ de la représentation (5.6)-(5.9) permet d'obtenir la formulation suivante du problème $\widetilde{\Gamma}(M, \Lambda')$:

$$-\sum_{i=1}^{r} d_i \mu_i \to \max \tag{5.11}$$

sous les conditions:

$$\sum_{i=1}^{\gamma_1} \alpha_i (-B^{(0)} + A^{(0)} X_i) + \sum_{i=\gamma_1+1}^{\gamma} \alpha_i A^{(0)} X_i - \sum_{i=1}^{r} \mu_i D_i = 0; \tag{5.12}$$

$$\sum_{i=1}^{\gamma_1} \alpha_i = 1; \tag{5.13}$$

$$\alpha_i \geqslant 0, \quad i=1, 2, \ldots, \gamma; \quad \mu_i \geqslant 0, \quad i=1, 2, \ldots, r. \tag{5.14}$$

Ici $D_i = (d_{i1}, d_{i2}, \ldots, d_{im})^T$, $i = 1, 2, \ldots, r$.

Etant donné que X_i, $i = 1, 2, \ldots, \gamma$, est une collection complète des sommets et des vecteurs directeurs des arêtes non bornées de l'ensemble $G_{\Lambda'}$, les conditions (5.12)-(5.14) peuvent être mises sous la forme équivalente

$$A^{(0)}X - \sum_{i=1}^{r} \mu_i D_i = B^{(0)}, \tag{5.15}$$

$$X \in G_{\Lambda'}, \tag{5.16}$$

$$\mu_i \geqslant 0, \quad i = 1, 2, \ldots, r. \tag{5.17}$$

Déterminons sur l'ensemble $G_{\Lambda'}$ la fonction

$$\varphi_2(X) = \inf_{l \in M} (B^{(0)} - A^{(0)}X, l). \tag{5.18}$$

Par définition, $\varphi_2(X)$ est la valeur extrémale de la forme linéaire dans le problème de minimisation de

$$(B^{(0)} - A^{(0)}X, l)$$

sous les conditions (5.9). Le problème dual au problème donné s'écrit

$$-\sum_{i=1}^{r} d_i \mu_i \to \max \tag{5.19}$$

sous les conditions

$$-\sum_{i=1}^{r} \mu_i D_i = B^{(0)} - A^{(0)}X, \tag{5.20}$$

$$\mu_i \geqslant 0, \quad i = 1, 2, \ldots, r. \tag{5.21}$$

Les conditions (5.9) étant compatibles, il vient

$$\varphi_2(X) = \begin{cases} -\infty \text{ si la solution du problème (5.19)-(5.21) est irréalisable;} \\ -\sum_{i=1}^{r} d_i \mu_i^*(X) \text{ si } (\mu_1^*(X), \ldots, \mu_r^*(X)) \text{ est solution du problème (5.19)-(5.21).} \end{cases}$$

Par suite, le problème $\widetilde{\Gamma}(M, \Lambda')$ peut s'écrire également sous une forme plus symétrique

$$\varphi_2(X) \to \max$$

sous la condition

$$X \in G_{\Lambda'}.$$

L'existence d'une solution réalisable du problème $\widetilde{\Gamma}(M, \Lambda')$ résulte de la résolubilité du problème $\Gamma(M, \Lambda')$.

5.2. Supposons que les solutions l^* et X^* des problèmes $\Gamma(M, \Lambda')$

et $\widetilde{\Gamma}(M, \Lambda')$ aient été trouvées. D'après l'inégalité (5.10), deux cas peuvent alors se présenter:

1) $v_0 = \varphi_1(l^*) = \varphi_2(X^*) = 0$;

2) $v_0 < 0$.

La première éventualité indique que Λ' est le point cherché dans lequel la fonction $f(\Lambda)$ atteint son minimum. De plus, X^* est la solution du problème étudié (3.1)-(3.4) décomposé en blocs. En effet, d'après (5.3), pour un $\varepsilon > 0$ suffisamment petit, le vecteur

$$l_\varepsilon = \varepsilon (A^{(0)}X^* - B^{(0)}) \in M.$$

C'est pourquoi

$$0 = v_0 = \varphi_2(X^*) \leqslant (B^{(0)} - A^{(0)}X^*, l_\varepsilon) = -\varepsilon |B^{(0)} - A^{(0)}X^*|^2,$$

soit

$$B^{(0)} = A^{(0)}X^*.$$

En outre, $X^* \in G_{\Lambda'}$. Notre proposition découle alors des conditions d'optimalité (3.36), (3.37) du programme d'un problème décomposé en blocs. Supposons maintenant qu'ait lieu le cas 2°. Puisque $\varphi_1(l^*) < 0$, le vecteur l^* est une direction possible issue du point Λ'.

Ainsi, le choix d'une direction possible issue du point Λ' se ramène à la résolution des problèmes duals $\Gamma(M, \Lambda')$ et $\widetilde{\Gamma}(M, \Lambda')$, l'ensemble convexe M ne devant être soumis qu'aux restrictions (5.3) et (5.4). En faisant varier les ensembles M et les méthodes d'exploration des problèmes $\Gamma(M, \Lambda')$ et $\widetilde{\Gamma}(M, \Lambda')$, on obtient différents procédés T du choix de la direction possible. Le plus commode est de former un ensemble M en partant des contraintes les plus simples, du type

$$l_i \leqslant 1, \quad l_i \geqslant -1.$$

Dans ce cas les vecteurs D_i sont au signe près des vecteurs unités.

5.3. L'une des méthodes possibles de résolution des problèmes $\Gamma(M, \Lambda')$ et $\widetilde{\Gamma}(M, \Lambda')$ consiste à minimiser directement la fonction $\varphi_1(l)$ avec $l \in M$. La réalisation de cette méthode ne diffère guère de la résolution du problème décomposé en blocs (3.1)-(3.4) décrite au § 4. La seule différence est que dans le cas considéré $C = 0$, $G = G_{\Lambda'}$ et, outre les hyperplans engendrés par les sommets et les arêtes infinies de l'ensemble $G_{\Lambda'}$, il faut tenir compte des hyperplans

$$\sum_{j=1}^{m} d_{ij} l_j = d_i$$

qui correspondent à l'ensemble M.

En utilisant cette méthode, on aura intérêt à choisir pour M un polyèdre

$$|l_i| \leqslant 1, \quad i = 1, 2, \ldots, m,$$

tel qu'il vérifie, apparement, les conditions (5.3), (5.4). Dans l'hy-

pothèse que ensemble $G_{\Lambda'}$ soit borné, la fonction $\varphi_1(l)$ est définie pour tout l. Par conséquent, un sommet arbitraire de M, $l^{(0)} = (-1, -1, \ldots, -1)$ par exemple, peut être pris comme point de départ de la minimisation de la fonction $\varphi_1(l)$. Dans le cas présent, la matrice initiale (4.4) a la forme

$$(e_1, e_2, \ldots, e_m, Q)^{-1},$$

où e_i est le vecteur unité de l'espace de dimension $(m+1)$ avec l'unité à la i-ème place;
$Q = (A^{(0)}X_0 - B^{(0)}, 1)^T$;
X_0 l'un des sommets de $G_{\Lambda'}$ sur lequel on obtient le maximum de la forme linéaire

$$-(l^{(0)}A^{(0)}, X) \quad \text{avec} \quad X \in G_{\Lambda'}.$$

5.4. Une autre méthode d'exploration des problèmes $\Gamma(M, \Lambda')$ et $\bar{\Gamma}(M, \Lambda')$ est associée à la maximisation de la fonction $\varphi_2(X)$. Examinons-la de plus près.

Soit $I(\Lambda')$ l'ensemble des numéros ν de tous ceux des vecteurs X_ν qui participent à la formation de l'ensemble $G_{\Lambda'}$. L'écriture adoptée pour le problème (3.5)-(3.7) permet de mettre le problème (5.11)-(5.14) sous la forme équivalente suivante

$$-\sum_{i=1}^{r} d_i\mu_i \to \max \tag{5.22}$$

sous les conditions

$$\sum_{\nu \in I(\Lambda')} z_\nu \bar{P}_\nu - \sum_{i=1}^{r} \mu_i \bar{D}_i = \bar{B}^{(0)}, \tag{5.23}$$

$$z_\nu \geqslant 0, \quad \mu_i \geqslant 0. \tag{5.24}$$

Ici $\bar{D}_i = \binom{D_i}{0}$.

Supposons que nous connaissons l'un des points $X = \sum_{\nu \in I(\Lambda')} z'_\nu X_\nu$ de l'ensemble $G_{\Lambda'}$ auquel correspond le vecteur

$$\bar{P} = \sum_{\nu \in I(\Lambda')} z'_\nu \bar{P}_\nu.$$

Calculons le vecteur $\bar{P} - \bar{B}^{(0)}$. Posons

$$\bar{D}_i = (-1)^{\delta_i} e_i, \quad i = 1, 2, \ldots, r = m,$$

où e_i est le vecteur unité avec l'unité à la i-ème place;

$$\delta_i = \begin{cases} 0 \text{ si la } i\text{-ème composante du vecteur } \bar{P} - \bar{B}^{(0)} \\ \quad \text{est non négative;} \\ 1 \text{ si la } i\text{-ème composante du vecteur } \bar{P} - \bar{B}^{(0)} \\ \quad \text{est négative.} \end{cases}$$

Il est évident que pour un pareil choix des vecteurs $\overline{D}_i$ les conditions (5.23), (5.24) sont compatibles, c'est-à-dire que l'ensemble convexe M vérifie la restriction (5.4). En outre, si pour tout i $d_i = 1$, la condition (5.3) est également respectée. Dans ce cas les conditions (5.23) se transforment en un système d'équations

$$\sum_{\nu \in I(\Lambda')} z_\nu \overline{P}_\nu - \sum_{i=1}^{m} \mu_i \overline{D}_i = \overline{B}^{(0)}. \tag{5.25}$$

Ainsi, la résolution du problème $\widetilde{\Gamma}(M, \Lambda')$, l'ensemble M étant choisi, se ramène à la minimisation de la forme linéaire

$$\sum_{i=1}^{m} \mu_i \tag{5.26}$$

sous les conditions (5.24) et (5.25).

Supposons qu'avec un certain ν_0 $\overline{P} = \overline{P}_{\nu_0}$, c'est-à-dire que le point $X = X_{\nu_0}$ soit un sommet de l'ensemble $G_{\Lambda'}$. Les vecteurs

$$\overline{P}_{\nu_0}, -\overline{D}_1, -\overline{D}_2, \ldots, -\overline{D}_m$$

constituent la base du programme de base du problème (5.24)-(5.26), qui peut être prise comme base initiale. Il est clair que la matrice inverse de la matrice de cette base se calcule sans peine.

Comme nous le verrons plus bas, on détermine parfois, au terme des étapes antérieures de la résolution d'un problème décomposé en blocs, un vecteur $\overline{P}$ tel que

$$\overline{P} = \sum_{\alpha=1}^{t} z'_{\nu_\alpha} \overline{P}_{\nu_\alpha}, \quad \nu_\alpha \in I(\Lambda'),$$

$$\overline{P} - \overline{B}^{(0)} = \sum_{\alpha=1}^{m+1-t} \mu'_{i_\alpha} \overline{D}_{i_\alpha},$$

les vecteurs

$$\overline{P}_{\nu_1}, \ldots, \overline{P}_{\nu_t}, \overline{D}_{i_1}, \ldots, \overline{D}_{i_{m+1-t}}$$

constituant la base du programme du problème (5.24)-(5.26) avec une matrice inverse connue. Une telle base peut aussi naturellement être prise pour base initiale.

5.5. Après ces remarques préalables relatives au choix rationnel d'un polyèdre M, qui rend plus facile la recherche de la base initiale du problème (5.24)-(5.26), passons à la description de la procédure de résolution. La discussion du problème (5.24)-(5.26) se fonde sur la méthode de décomposition et compte plusieurs itérations.

Supposons que nous ayons une certaine base de ce problème, ainsi que le vecteur $\overline{l}' = (l', l'_0)$ des critères de sélection de ses conditions par rapport à la base choisie.

Il se peut que le programme de base du problème (5.24)-(5.26) associé à cette base rend nulle sa forme linéaire (5.26). Cette éventualité témoigne de l'absence de directions possibles issues du point Λ'. En même temps a lieu l'égalité

$$\overline{B}^{(0)} = \sum_{i=1}^{t} z'_{\nu_i} \overline{P}_{\nu_i}, \quad t \leqslant m+1,$$

où les z'_{ν_i} sont les composantes de base non nulles du programme de base donné du problème (5.24)-(5.26).

Donc, le vecteur

$$X = \sum_{i=1}^{t} z'_{\nu_i} X_{\nu_i}$$

est la solution cherchée du problème décomposé en blocs (3.1)-(3.4). Supposons maintenant que la forme linéaire (5.26) prenne sur le programme examiné une valeur positive.

Vérifions d'abord si les inégalités

$$-(D_i, \ l') \leqslant d_i$$

sont respectées ou, ce qui revient au même, si

$$1 + (-1)^{\delta_i} l'_i \geqslant 0, \quad i = 1, 2, \ldots, m. \tag{5.27}$$

Si l'une d'entre elles au moins n'est pas satisfaite, le vecteur correspondant

$$-D_i = -(-1)^{\delta_i} e_i$$

est inclus dans la base. Dans le cas où toutes les inégalités (5.27) sont observées, on passe à la formation de ce qu'on appelle le problème auxiliaire engendré par le vecteur $\overline{l}'$.

Les conditions du problème auxiliaire sont déterminées par l'ensemble $G_{\Lambda'}$. Soit X' un sommet de l'ensemble $G_{\Lambda'}$, c'est-à-dire le programme de base optimal d'un $X_{\Lambda'}$-problème:

$$(C_{\Lambda'}, \ X) \to \max, \tag{5.28}$$

$$\sum_{j=1}^{n} A_j^{(1)} x_j = A^{(1)} X = B^{(1)}, \tag{5.29}$$

$$X \geqslant 0. \tag{5.30}$$

Si $\Delta'_j \geqslant 0$ est le critère de sélection du j-ième vecteur des conditions du problème (5.28)-(5.30) par rapport à la base du programme X', l'ensemble $G_{\Lambda'}$ est défini par les conditions (5.29), (5.30) et par la restriction supplémentaire

$$x_j = 0 \quad \text{si} \quad \Delta'_j > 0. \tag{5.31}$$

Par suite, le système de conditions qui définit l'ensemble convexe $G_{\Lambda'}$ s'écrit

$$\sum_{j \in J(\Lambda')} A_j^{(1)} x_j = B^{(1)}, \tag{5.32}$$

$$x_j \geqslant 0, \quad j \in J(\Lambda'), \tag{5.33}$$

où $J(\Lambda')$ est l'ensemble des indices j tels que $\Lambda'_j = 0$.

Le problème auxiliaire engendré par le vecteur $\bar{l}' = (l', l'_0)$ consiste à maximiser la forme linéaire

$$(l'A^{(0)}, X) \tag{5.34}$$

tout en respectant les conditions (5.32), (5.33).

Remarquons que le nombre de variables du problème (5.32)-(5.34) dépasse généralement de m au plus le nombre de ses contraintes. Supposons que le problème auxiliaire soit résolu et que X'' en soit solution de base (pour plus de concision, nous omettons le cas où la forme linéaire (5.34) est infinie sur l'ensemble $G_{\Lambda'}$).

Si

$$(l'A^{(0)}, X'') + l'_0 = 0,$$

$-l'$ est une direction possible, puisque

$$\varphi_1(-l') = \max_{X \in G_{\Lambda'}} (l'A^{(0)}, X) - (B^{(0)}, l') = (l'A^{(0)}, X'') - (B^{(0)}, l') =$$

$$= -l'_0 - (B^{(0)}, l') = -\sum_{i=1}^{m} \mu'_i < 0.$$

L'analyse des problèmes $\Gamma(M, \Lambda')$ et $\tilde{\Gamma}(M, \Lambda')$ est ainsi terminée.

Mais si

$$(l'A^{(0)}, X'') + l'_0 > 0,$$

il faut poursuivre la procédure de résolution du problème (5.24)-(5.26). On introduit dans la base disponible de ce problème le vecteur $\bar{P}_{\nu_k}$ défini par le sommet trouvé $X_{\nu_k} = X''$ de l'ensemble $G_{\Lambda'}$. Après avoir obtenu la nouvelle base et le vecteur des critères $\bar{l}'' = (l'', l''_0)$ des conditions du problème qui lui correspond, reprenons la suite d'opérations décrite dans ce qui précède. Finalement la procédure aboutit à la construction d'une direction possible, ou bien révèle que Λ' est le point du minimum de la fonction $f(\Lambda)$ définie sur R. Dans ce dernier cas, la solution cherchée du problème décomposé en blocs (3.1)-(3.4) s'obtient automatiquement.

5.6. Les modes du choix des directions possibles décrits aux paragraphes 5.3 à 5.5. sont associés soit à la minimisation de la fonction $\varphi_1(l)$ sur l'ensemble M, soit à la maximisation de la fonction $\varphi_2(X)$ sur l'ensemble $G_{\Lambda'}$. Ils sont tous fondés sur des méthodes finies de programmation linéaire. Examinons encore un mode de

choix des directions possibles, qui se base sur une procédure itérative de résolution des jeux proposée par Brown (cf. § 3 du chapitre 2, où cette méthode porte le nom de jeu fictif). Pour utiliser cette méthode, on part de vecteurs arbitraires $l^{(1)} \in M$ et $X^{(1)} \in G_{\Lambda'}$ et l'on obtient, en règle générale, des suites infinies de vecteurs convergeant vers les solutions des problèmes $\Gamma(M, \Lambda')$ et $\widetilde{\Gamma}(M, \Lambda')$.

Soient $l^{(k)}$ et $X^{(k)}$ les k-ièmes éléments des suites considérées. Décrivons l'ordre des opérations nécessaires pour calculer les vecteurs

$$X^{(k+1)} \quad \text{et} \quad l^{(k+1)},$$

en supposant que les ensembles M et $G_{\Lambda'}$ soient bornés.

Considérons le problème de programmation linéaire

$$(l^{(k)} A^{(0)}, X) \to \min \tag{5.35}$$

avec les conditions

$$X \in G_{\Lambda'}, \tag{5.36}$$

dont l'écriture algébrique est du type (5.32)-(5.33). Soit $\widetilde{X}^{(k+1)}$ une solution de base quelconque du problème (5.35), (5.36). Posons

$$X^{(k+1)} = \frac{1}{k+1} \widetilde{X}^{(k+1)} + \frac{k}{k+1} X^{(k)}. \tag{5.37}$$

Pour définir la nouvelle direction $l^{(k+1)}$, cherchons la solution du problème de programmation linéaire

$$(B^{(0)} - A^{(0)} X^{(k+1)}, l) \to \min \tag{5.38}$$

sous les conditions

$$l \in M. \tag{5.39}$$

Si $\widetilde{l}^{(k+1)}$ est un programme de base optimal du problème (5.38), (5.39), posons

$$l^{(k+1)} = \frac{1}{k+1} \widetilde{l}^{(k+1)} + \frac{k}{k+1} l^{(k)}. \tag{5.40}$$

Le polyèdre M doit être choisi de structure la plus simple possible. On peut prendre pour M, par exemple, un cube de dimension m :

$$|l_i| \leqslant 1, \quad i = 1, 2, \ldots, m. \tag{5.41}$$

Il est évident que la solution du problème (5.38), (5.39) pour le cas où M est de la forme (5.41) s'obtient sans peine. Aussi, l'essentiel du travail nécessaire pour l'exécution d'une étape consiste-t-il à réaliser l'exploration du problème (5.35), (5.36).

On voit sans peine que l'hypothèse sur l'allure bornée des ensembles $G_{\Lambda'}$ et M justifie les relations

$$\left.\begin{aligned} \lim_{k\to\infty} \varphi_1(l^{(k)}) &= \min_{l \in M} \varphi_1(l), \\ \lim_{k\to\infty} \varphi_2(X^{(k)}) &= \max_{X \in G_{\Lambda'}} \varphi_2(X). \end{aligned}\right\} \tag{5.42}$$

Pour cela, il faut réduire à un jeu rectangulaire le problème de la recherche du point selle de la fonction $(B^{(0)} - A^{(0)}X, l)$ avec $X \in G_{\Lambda'}$, $l \in M$, après avoir représenté les polyèdres $G_{\Lambda'}$ et M comme les ensembles des combinaisons convexes de leurs sommets, puis se servir du résultat de J. Robinson établissant la convergence de la méthode de Brown (cf. § 3 du chapitre 2). La démonstration détaillée du bien-fondé des relations (5.42) est laissée au lecteur (cf. exercice 12).

On sait de la pratique de calcul fournie par l'application de la méthode de Brown que cette méthode permet d'obtenir assez vite une approximation grossière des stratégies optimales, après quoi généralement sa convergence ralentit brusquement. Cette particularité, propre à toutes les méthodes itératives connues de résolution des jeux, justifie la recommandation qui suit. Il est rationnel d'utiliser la méthode itérative du choix des directions possibles décrite ci-dessus au début du processus de résolution du problème décomposé en blocs, lorsque la quantité $\min\limits_{l \in M} \varphi_1(l) < 0$ est suffisamment petite. C'est le cas où, pour des valeurs de k relativement faibles, la quantité $\varphi_1(l^{(k)})$ devient négative et la direction $l^{(k)}$ possible. L'avantage de la méthode itérative par rapport aux méthodes finies du choix des directions possibles consiste dans le fait qu'elle rend inutiles la composition et la transformation de la matrice d'ordre $m + 1$ engendrée par les vecteurs $\overline{P}_\nu$ et $\overline{D}_i$. Cependant une fois que le point Λ' se trouve dans un certain voisinage du point du minimum cherché de la fonction $f(\Lambda)$, l'avantage mentionné de la méthode itérative ne compense déjà plus le net ralentissement de sa convergence.

On voit ainsi que c'est l'utilisation commune des procédés finis et itératifs du choix des directions possibles qui apparaît comme la plus efficace. Au début de la procédure de résolution d'un problème décomposé en blocs il convient de donner la préférence au procédé itératif. Lorsqu'on aura décelé un ralentissement de la convergence de ce procédé, il sera rationnel de passer à l'un des procédés finis exposés aux paragraphes 5.3-5.5.

Appelons *grande itération* de la programmation par blocs l'ensemble des opérations nécessaires pour passer du point $\Lambda' \in R$ à un nouveau point $\Lambda'' \in R$. Nous avons examiné quelques réalisations de la première étape de la grande itération, qui consiste à choisir une direction possible l'.

Chacune de ces réalisations nécessite la résolution de plusieurs problèmes auxiliaires de programmation linéaire sous les conditions (5.29)-(5.31). Ces dernières ne se distinguent des conditions (3.3), (3.4) que par une partie (généralement la plus grande) des variables x_j posée d'avance égale à zéro.

5.7. Passons à la description de la deuxième étape de la grande itération qui consiste à choisir la valeur du déplacement le long de la direction possible.

Donc, soit l' la direction possible issue du point Λ'. A la deuxième étape, le point Λ' est remplacé par

$$\Lambda'' = \Lambda(\theta') = \Lambda' + \theta' l'.$$

Le choix du paramètre θ' se fait d'après l'algorithme de programmation paramétrique exposé au § 1 du chapitre 3.

Considérons le problème paramétrique qui consiste à maximiser la forme linéaire

$$(C - (\Lambda' + \theta l') A^{(0)}, X) = (C' + \theta C'', X) \tag{5.43}$$

sous les conditions

$$X \in G. \tag{5.44}$$

Soit

$$g(\theta) = \sup_{X \in G} (C' + \theta C'', X) + \theta (l', B^{(0)}). \tag{5.45}$$

Il est évident que

$$f(\Lambda' + \theta l') = g(\theta) + (\Lambda', B^{(0)}).$$

$g(\theta)$ est une fonction linéaire par morceaux convexe vers le bas; de plus, par suite du choix de la direction l' lorsque les valeurs positives de l'argument sont suffisamment petites, $g(\theta)$ décroît avec l'augmentation de θ.

Pour étudier le problème (5.43), (5.44) avec $\theta \geqslant 0$, nous appliquerons la méthode de programmation paramétrique. La procédure de l'exploration peut conduire à l'un des résultats suivants.

1°. On obtient la valeur critique maximale $\theta' > 0$, à droite de laquelle la fonction $g(\theta)$ ne décroît plus.

2°. On obtient le programme de base $\widetilde{X}$, optimal pour le problème (5.43), (5.44) lorsque $\theta \geqslant \theta_1$, et de plus,

$$(C'', \widetilde{X}) + (l', B^{(0)}) < 0,$$

c'est-à-dire que $g(\theta)$ décroît sur tout le demi-axe positif $\theta \geqslant 0$.

Dans le premier cas, on se déplace en le point

$$\Lambda'' = \Lambda' + \theta' l'$$

pour passer à l'itération suivante. Notons que la valeur positive de θ' justifie l'inégalité

$$f(\Lambda'') < f(\Lambda'). \tag{5.46}$$

Dans le deuxième cas, on conclut que les conditions (3.2)-(3.4) du problème sont incompatibles, étant donné que

$$\inf_{\Lambda \in R} f(\Lambda) \leqslant \inf_{\theta \geqslant 0} g(\theta) = -\infty.$$

Désignons par $E_{\min}$ le mode du choix de θ' associé à la minimisation de la fonction $g(\theta)$.

L'utilisation du mode $E_{\min}$ est rationnelle dans les itérations du stade initial de la procédure, lorsque nous sommes encore assez loin du minimum de la fonction $f(\Lambda)$. A ce stade l'avantage du procédé $E_{\min}$ qui consiste en une décroissance assez rapide de la fonction $f(\Lambda)$ à chaque itération, compense son inconvénient — la complexité d'emploi de l'information accumulée lors du choix de la direction possible à l'itération précédente. Cet inconvénient se manifeste dans le cas où le choix de la direction possible s'effectue à l'aide de l'un des procédés finis T. Lorsqu'on utilise le procédé $E_{\min}$, la base initiale du problème de choix de la direction possible est formée à nouveau (elle contient m vecteurs unitaires), ce qui ralentit la procédure de recherche de cette direction.

A mesure qu'on s'approche du minimum cherché de la fonction $f(\Lambda)$, la valeur de son décroissement lors du passage de Λ' à Λ'' diminue nettement. En outre, au stade final de la procédure, il est avantageux de refaire plus souvent le calcul de la direction possible. Aussi est-il particulièrement important d'utiliser ici un mode de choix de θ' tel qui, sans minimiser $f(\Lambda)$ le long de la direction choisie, fournit néanmoins une bonne première approximation pour le problème successif de la recherche d'une direction possible. Le procédé E_0, dont voici la description, satisfait à ces exigences.

On adopte pour θ' la valeur critique positive minimale du paramètre θ du problème (5.43), (5.44) (si elle existe bien sûr).

Mais si dans le problème (5.43), (5.44) le paramètre l n'a pas de valeurs critiques sur la demi-droite $(0, \infty)$, c'est-à-dire si $g(\theta)$ est une fonction linéaire décroissante de θ sur le demi-axe positif, on fixe que les conditions du problème (3.1)-(3.4) sont incompatibles.

Supposons que la base finale du problème (5.22)-(5.24) de choix de la direction possible pour le point Λ' comporte les vecteurs $\overline{P}_{s_1}, \overline{P}_{s_2}, \ldots, \overline{P}_{s_d}$ et que θ' soit la première valeur critique positive du paramètre θ. Ceci signifie que les points $\overline{\Lambda}(\theta) = (\Lambda' + \theta l', f(\Lambda' + \theta l'))$, $0 \leqslant \theta \leqslant \theta'$, situés à la surface (3.25), appartiennent aux hyperplans H_{s_i}, $i = 1, 2, \ldots, d$, engendrés par les vecteurs $X_{s_1}, X_{s_2}, \ldots, X_{s_d}$ (sommets ou vecteurs directeurs des arêtes infinies de l'ensemble $G_{\Lambda'}$). C'est pourquoi $\overline{P}_{s_1}, \ldots, \overline{P}_{s_d}$ constituent les vecteurs des conditions du problème (5.22)-(5.24) non seulement pour le point Λ', mais encore pour le point Λ''. Autrement dit, la base finale du problème (5.22)-(5.24) pour le point Λ' peut être prise comme base initiale de ce problème pour le point Λ''. Insistons sur le fait que les considérations ci-dessus concernent les procédés de choix de la direction possible, fondés sur la maximisation de la fonction $\varphi_2(X)$. La possibilité de la mise en œuvre de toute

l'information accumulée aux itérations précédentes est un mérite important de ses procédés.

Soit X_ν le sommet de l'ensemble $G_{\Lambda'}$ qui a engendré le vecteur $\overline{P}_\nu$ introduit dans la base optimale du problème (5.22)-(5.24). Il est évident que X_ν est solution du problème (5.43), (5.44) pour $0 \leqslant \theta \leqslant \theta'$. Par conséquent, dans le cas non dégénéré la valeur cherchée de θ' s'obtient dès la première étape de la discussion du problème (5.43), (5.44), si seulement l'on adopte X_ν comme un programme initial. Il est commode de prendre pour X_ν le programme optimal du dernier problème auxiliaire formé au cours du choix de la direction possible.

5.8. Les modes examinés de choix d'une direction possible et de détermination de la valeur du déplacement le long de cette direction engendrent toute une classe de méthodes de programmation par blocs. Nous avons déjà dit que le plus efficace est de combiner les différents procédés T et E au sein d'une même procédure. Au début de la résolution d'un problème décomposé en blocs, il est rationnel d'appliquer le procédé itératif du choix de la direction possible l' (paragraphe 4.6) et de se déplacer le long de l' tant que la fonction à minimiser décroît (mode $E_{\min}$). A mesure qu'on s'approche du minimum cherché de la fonction $f(\Lambda)$, la convergence de la méthode itérative devient généralement plus lente. Ceci une fois établi, il convient de passer à l'une des méthodes finies. Au dernier stade de la résolution mieux vaut déterminer la longueur de l'étape θ' par le procédé E_0, c'est-à-dire se déplacer jusqu'à la première valeur critique du problème paramétrique correspondant. Lorsqu'on utilise le procédé E_0, le plus commode est de choisir la direction possible par maximisation de la fonction $\varphi_2(X)$. Nous avons déjà dit que dans ce cas on peut prendre comme programme initial du problème de choix d'une direction possible pour le point Λ'' le programme optimal du problème analogue pour le point précédent Λ'.

Le schéma proposé se distingue des procédés usuels de programmation linéaire par une plus grande souplesse, ce qui rend possible son adaptation immédiate en cours de résolution aux particularités d'un problème donné (ou aux particularités des problèmes de la classe donnée).

Logiquement plus simples, bien que moins efficaces, sont les schémas de programmation par blocs qui utilisent dans toute la procédure de résolution des modes T et E fixés. Tels sont, par exemple, l'analogue dual de la méthode de décomposition (§ 4), dans lequel les directions possibles sont choisies parmi les arêtes issues d'un certain sommet de l'ensemble convexe $\overline{R}$. Dans cette méthode, la longueur de l'étape est recherchée à l'aide du procédé E_0. Lors de l'utilisation de l'analogue dual de la méthode de décomposition, le Z-problème (3.5)-(3.7) est résolu par la méthode duale du simplexe.

Si la direction possible l est choisie par maximisation de la

fonction $\varphi_2(X)$, l'ensemble M étant défini par les conditions les plus simples, du type

$$l_i \geqslant -1, \quad i = 1, 2, \ldots, m,$$

et la longueur de l'étape déterminée par le mode E_0, on aboutit à la méthode de programmation par blocs analogue de la méthode primale-duale.

Nous laissons au lecteur le soin de vérifier que la méthode considérée est liée à la résolution du Z-problème (3.5)-(3.7) par la méthode primale-duale (cf. exercice 13).

§ 6. Choix de l'approximation initiale

6.1. Un des avantages des méthodes de programmation par blocs fondées sur la minimisation de la fonction $f(\Lambda)$ (§ 5) est la simplicité relative du choix de l'approximation initiale.

En effet, dans ces méthodes on peut adopter comme approximation initiale n'importe quelle valeur de Λ telle que la forme linéaire

$$(C_\Lambda, X) = (C - \Lambda A^{(0)}, X)$$

soit bornée sur l'ensemble G.

Lorsque l'ensemble G est borné (cas le plus typique dans les problèmes pratiques), on peut prendre évidemment pour Λ un point quelconque de l'espace de dimension m. Une possibilité aussi large de choix de l'approximation initiale permet de profiter de toutes sortes de considérations supplémentaires concernant le point initial Λ et portant sur le sens physique (économique) du problème pratique à étudier.

Lorsque l'ensemble G est non borné, il est rationnel d'introduire dans le système des conditions (3.3) la contrainte supplémentaire

$$\sum_{j=1}^{n} x_j \leqslant Q, \tag{6.1}$$

où Q est un nombre suffisamment grand.

Le choix du point initial n'impose pas alors de calculs. Il faut conserver la contrainte (6.1) tant qu'on n'obtienne au cours de la procédure de résolution du problème (3.1)-(3.4) le point Λ' tel que $f(\Lambda')$ ne dépend pas de Q. Si, par contre, la valeur minimale de la fonction $f(\Lambda)$ dépend de Q, la forme linéaire (C_Λ, X) n'est pas bornée sur G quel que soit Λ, ce qui signifie que R est un ensemble vide. Dans ce cas le problème (3.1)-(3.4) n'a pas de solution. La justification de cette méthode peut se faire en partant de la méthode connue de construction d'un programme initial du problème dual (cf. chapitre 9, paragraphe 5.4 de [87]). Nous laissons au lecteur le soin de réaliser cette tâche (cf. exercice 14).

6.2. La méthode de résolution d'un problème décomposé en blocs, fondée sur le déplacement suivant les sommets de l'ensemble convexe $\overline{R}$ (§ 4), suppose connu un sommet initial de cet ensemble.

La procédure de passage d'un point arbitraire de la surface (3.25) à l'un de ses sommets s'établit aisément sur la base de ce que l'on appelle la méthode du gradient (cf. chapitre 9, paragraphe 5.3 de [87]).

Voici la description de cette procédure. Soit Λ' un point quelconque de l'ensemble R. Choisissons l'un des sommets de l'ensemble $G_{\Lambda'}$ que nous allons noter par X_1. Au sommet X_1 correspond l'hyperplan

$$\lambda_{m+1} + (A^{(0)} X_1 - B^{(0)}, \Lambda) = (C, X_1), \tag{6.2}$$

qui passe par le point $\overline{\Lambda}' = (\Lambda', f(\Lambda'))$ de la surface polyédrique

$$\lambda_{m+1} = f(\Lambda). \tag{6.3}$$

Cherchons, en partant du point $\overline{\Lambda}'$, à progresser suivant l'hyperplan (6.2) dans la direction

$$\overline{l}' = (l', -l'_{m+1}), \quad l' = A^{(0)} X_1 - B^{(0)},$$
$$l'_{m+1} = |A^{(0)} X_1 - B^{(0)}|^2 = (l', l'),$$

qui correspond à la plus grande vitesse de décroissance de la $(m+1)$-ième coordonnée. Ce déplacement correspond au passage du point $\overline{\Lambda}' = (\Lambda', f(\Lambda'))$ au point

$$\overline{\Lambda}(\theta) = (\Lambda' + l'\theta, f(\Lambda') - l'_{m+1}\theta), \quad \theta \geqslant 0.$$

Le paramètre θ est choisi le plus grand possible, mais de façon à vérifier l'égalité

$$f(\Lambda' + l'\theta) = f(\Lambda') - l'_{m+1}\theta. \tag{6.4}$$

Désignons par θ' la valeur correspondante du paramètre θ. Avec $0 \leqslant \theta \leqslant \theta'$, le point $\overline{\Lambda}(\theta)$ reste sur la surface (6.3). Lorsque $\theta > \theta'$, le point $\overline{\Lambda}'(\theta)$ se trouve au-dessous de cette surface. Le choix de la quantité θ' se fait d'après la méthode de programmation paramétrique. Le problème exploré est, notamment,

$$(C_{\Lambda'} - \theta l' A^{(0)}, X) \to \max \tag{6.5}$$

$$\text{avec } X \in G. \tag{6.6}$$

Si pour tout $\theta > 0$ la forme linéaire (6.5) est non bornée sur G, on pose $\theta' = 0$, et l'on désigne par X_2 le vecteur directeur de l'arête non bornée suivant laquelle la forme linéaire (6.5) tend vers l'infini. S'il n'en est pas ainsi, on cherche le sommet X' de l'ensemble G qui correspond au programme optimal du problème (6.5), (6.6) sur un certain segment $[0, \theta_1]$, $\theta_1 > 0$, et l'on vérifie si l'égalité (6.4) est respectée.

Pour

$$(l'A^{(0)},\ X') - (l',\ B^{(0)}) = (l',\ A^{(0)}X' - B^{(0)}) < l'_{m+1},$$

posons

$$X_2 = X', \quad \theta' = 0.$$

Mais si

$$(l',\ A^{(0)}X' - B^{(0)}) = l'_{m+1},$$

c'est-à-dire si (6.4) se vérifie pour $\theta \in [0,\ \theta_1]$, on détermine la valeur critique positive minimale θ^* du paramètre du problème (6.5), (6.6), ou bien l'on constate l'absence de valeurs critiques sur la demi-droite $(0,\ \infty)$. Dans le premier cas $\theta' = \theta^*$, dans le deuxième, $\theta' = \infty$. Avec $\theta' = \infty$, la procédure prend fin, puisque

$$\inf_{\Lambda \in R} f(\Lambda) = -\infty,$$

c'est-à-dire la solution du problème (3.1)-(3.4) est irréalisable.

Mais si

$$\theta' = \theta^* < \infty,$$

on désigne alors par X_2 soit un sommet de l'ensemble G, qui constitue la solution du problème (6.5), (6.6) avec $\theta \in [\theta^*,\ \theta_2]$, $\theta_2 > 0$, soit le vecteur directeur d'une arête infinie de G le long de laquelle, lorsque $\theta > \theta^*$, la forme linéaire (6.5) tend vers l'infini.

Le vecteur X_2 définit l'hyperplan

$$\varepsilon\lambda_{m+1} + (A'^{(0)}X_2 - \varepsilon B^{(0)},\ \Lambda) = (C,\ X_2), \tag{6.7}$$

où ε est égal à l'unité ou à zéro suivant que X_2 est un sommet de l'ensemble G ou le vecteur directeur d'une arête non bornée de cet ensemble. Le point

$$\overline{\Lambda}'' = \overline{\Lambda}'(\theta')$$

appartient à l'intersection des hyperplans (6.2) et (6.7) qui sont évidemment linéairement indépendants.

6.3. Le déplacement ultérieur s'effectue dans la direction $\overline{l}''$ qui appartient aux deux hyperplans, $\overline{l}''$ étant choisie de façon que la $(m+1)$-ième coordonnée décroisse aussi vite que possible. A cet effet, les équations homogènes qui correspondent aux hyperplans (6.2), (6.7) et associent les composantes de la direction recherchée $\overline{l}''$ sont résolues par rapport aux inconnues λ_{m+1} et λ_k. Si l'on convient pour fixer les idées que $k = m$, on aboutit aux relations

$$\lambda_{m+1} + \sum_{i=1}^{m-1} \gamma_{i1}\lambda_i = 0, \quad \lambda_m + \sum_{i=1}^{m-1} \gamma_{i2}\lambda_i = 0. \tag{6.8}$$

Les premières $m-1$ composantes du vecteur $\overline{l}''$ coïncident avec les valeurs correspondantes de γ_{i1}; les deux dernières composantes sont déterminées d'après les équations (6.8).

Si $\overline{l}'' \neq 0$, on passe à la discussion du problème paramétrique

$$(C_{\Lambda''} - \theta l'' A^{(0)},\ X) \to \max$$

avec

$$X \in G.$$

En procédant exactement de la même façon que dans le cas du problème (6.5), (6.6), on constate que la fonction $f(\Lambda)$ n'est pas bornée inférieurement sur l'ensemble R, ou bien l'on passe du point $\overline{\Lambda}''$ au point $\overline{\Lambda}'''$. Le point $\overline{\Lambda}'''$ appartient à la surface (6.3) et se situe à l'intersection de trois hyperplans indépendants, dont deux sont définis par les équations (6.2) et (6.7) et le troisième est engendré par le vecteur X_2 obtenu lors de la construction du point $\overline{\Lambda}'''$.

6.4. Il se peut que $\overline{l}'' = 0$, c'est-à-dire que la partie commune des hyperplans (6.2) et (6.7) soit orthogonale au $(m+1)$-ième axe de coordonnées. Constatons que la direction $\overline{l}'$, choisie à la première étape, pouvait elle aussi être égale au vecteur nul. Mais cela aurait évidemment signifié que $A^{(0)}X_1 - B^{(0)} = 0$, c'est-à-dire que X_1 était un programme optimal du problème décomposé en blocs.

Soit $\overline{l}'' = 0$. Définissons la direçtion $l^{(k)} = (l_1^{(k)}, \ldots, l_{m+1}^{(k)})$ à l'aide de la relation

$$l_i^{(k)} = \begin{cases} 0\ , & i \neq k,\ 1 \leqslant i \leqslant m-1, \\ 1\ , & i = k, \\ -\gamma_{2k}, & i = m, \\ -\gamma_{1k} = 0, & i = m+1. \end{cases}$$
$$(k = 1, 2, \ldots, m-1),$$

où les paramètres γ_{1k} et γ_{2k} sont fournis par les équations (6.8).

Cherchons à nous déplacer du point Λ'' dans le sens de la droite parallèle à la direction $l^{(1)}$. A cette fin considérons le problème paramétrique

$$\left.\begin{aligned} (C_{\Lambda''} - \theta l^{(1)} A^{(0)},\ X) &\to \max, \\ X &\in G. \end{aligned}\right\} \tag{6.9}$$

Deux cas sont possibles :

a) le paramètre θ du problème (6.9) a des valeurs critiques sur la droite $(-\infty, \infty)$;

b) le paramètre θ de ce problème n'a aucune valeur critique.

Supposons qu'ait lieu le cas a).

Calculons la plus grande ou la plus petite valeur de θ telle que

$$\max_{X \in G} (G_{\Lambda''} - \theta l^{(1)} A^{(0)},\ X) + (B^{(0)},\ \theta l^{(1)}) = f(\Lambda'') - (B^{(0)},\ \Lambda'').$$

La valeur obtenue de $\theta = \theta''$ détermine, comme dans le cas précédent, le point $\overline{\Lambda}'''$ et le vecteur X_3 qui engendre un nouvel hyperplan contenant le point $\overline{\Lambda}'''$.

Ainsi, dans le cas a) il est également possible de passer à un point $\bar{\Lambda}'''$ qui appartient à l'intersection de trois hyperplans indépendants du type (6.7). Lors de ce passage, la $(m+1)$-ième coordonnée reste naturellement invariable.

Comme on le vérifie aisément (cf. exercice 15), le cas b) est équivalent à la justification de la condition

$$(l^{(1)}, B^{(0)} - A^{(0)}X) = 0 \quad \text{pour tout} \quad X \in G. \tag{6.10}$$

Cette dernière signifie que la première équation du système

$$A^{(0)}X = B^{(0)}$$

est une conséquence de sa m-ième équation et des relations

$$A^{(1)}X = B^{(1)}, \quad X \geqslant 0,$$

qui définissent l'ensemble G.

Par conséquent, la première ligne de la matrice $A^{(0)}$ peut être supprimée, ce qui diminue d'une unité la dimension de l'ensemble R.

Ceci fait, il faut remplacer la direction $l^{(1)}$ par $l^{(2)}$ et reprendre la procédure décrite. Dans le cas a) on passe au point $\bar{\Lambda}'''$ dont le nombre de composantes est m; mais si l'on est en présence du cas b), on supprime encore une ligne de la matrice $A^{(0)}$. Après avoir repris au besoin cette procédure plusieurs fois et passant successivement à $l^{(3)}$, $l^{(4)}$, etc., on obtient finalement le point $\bar{\Lambda}'''$ dont le nombre de composantes est $\leqslant m$ ou bien l'on supprime les premières $m-1$ lignes de la matrice $A^{(0)}$. Dans ce dernier cas, le point $(\lambda''_m, \lambda''_{m+1})$ est le sommet recherché de l'ensemble convexe $\bar{R}$ dont la dimension s'est trouvée être égale à deux.

6.5. Une fois obtenu le point $\bar{\Lambda}'''$ situé à l'intersection des trois hyperplans, on passe, du moins si le nombre des conditions restantes du système $A^{(0)}X = B^{(0)}$ dépasse trois, à l'étape suivante de la procédure. Cette étape, de même que celles qui la suivent, ne diffère en rien de la deuxième étape qui vient d'être examinée en détail.

Dans le cas général, $s \leqslant m$ étapes de la procédure fournissent le sommet recherché de l'ensemble $\bar{R}$ (la dimension de $\bar{R}$ décroît alors jusqu'à $s+1$), ou bien l'on constate que le problème considéré est irréalisable.

§ 7. Convergence des méthodes de programmation par blocs associées à la minimisation de la fonction $f(\Lambda)$

7.1. Nous avons déjà dit au paragraphe 5.8 que le plus avantageux est d'effectuer le stade final de la procédure de minimisation de la fonction $f(\Lambda)$ en se guidant par la méthode E_0 pour choisir

la grandeur du déplacement dans la direction possible. Montrons que dans ce cas la procédure de minimisation de la fonction $f(\Lambda)$ est finie.

Théorème 7.1. *Si, à partir d'un point $\Lambda^{(s)}$ la direction possible est choisie par minimisation de la fonction $\varphi_1(l)$ ou par maximisation de la fonction $\varphi_2(X)$ et si la longueur de l'étape dans la direction choisie est définie à l'aide du procédé E_0, on obtient après un nombre fini de grandes itérations le minimum de la fonction $f(\Lambda)$ et la solution du problème* (3.1)-(3.4), *ou bien l'on établit que les conditions* (3.2)-(3.4) *sont incompatibles.*

Démonstration. Considérons le système de conditions

$$\sum_{\nu=1}^{N} z_\nu \overline{P}_\nu - \sum_{i=1}^{r} \mu_i \overline{D}_i = \overline{B}^{(0)}, \quad z_\nu \geqslant 0, \ \mu_i \geqslant 0, \tag{7.1}$$

où, comme d'habitude, les $\overline{P}_\nu$ sont engendrés par les sommets et par les vecteurs directeurs des arêtes non bornées de l'ensemble G, alors que les vecteurs $\overline{D}_i = (D_i, 0)^T$ définissent l'ensemble M [cf. (5.5)]. Considérons, sans limiter la généralité, que ces conditions sont non dégénérées, c'est-à-dire supposons que $\overline{B}_0$ ne puisse pas être mis sous la forme d'une combinaison linéaire de moins de $m+1$ vecteurs $\overline{P}_\nu$ et $\overline{D}_i$. (On peut toujours y parvenir à l'aide de ε-méthode usuelle.) Ainsi qu'on l'a établi, le problème $\widetilde{\Gamma}(M, \Lambda')$ associé au point Λ' peut être mis sous la forme (5.22)-(5.24). Le problème $\Gamma(M, \Lambda')$ est dual au (5.22)-(5.24). Désignons par $S(M, \Lambda')$ la valeur extrémale commune des fonctions (5.1) et (5.18) des problèmes $\Gamma(M, \Lambda')$ et $\widetilde{\Gamma}(M, \Lambda')$.

Supposons que les vecteurs

$$\overline{P}_{\nu_1}, \overline{P}_{\nu_2}, \ldots, \overline{P}_{\nu_t}, -\overline{D}_{i_1}, -\overline{D}_{i_2}, \ldots, -\overline{D}_{i_{m+1-t}} \tag{7.2}$$

constituant la base du programme de base optimal du problème (5.22)-(5.24). La direction possible $\overline{l}'$ coïncidant avec le vecteur des critères des conditions du problème par rapport à la base optimale considérée, vérifie les relations

$$\begin{gathered} (\overline{l}', \overline{P}_{\nu_\alpha}) = 0, \ \alpha = 1, 2, \ldots, t, \\ (l', \overline{P}_\nu) \geqslant 0, \quad \nu \in I(\Lambda'), \end{gathered} \tag{7.3}$$

où l'ensemble des indices $I(\Lambda')$ est déterminé par la restriction

$$\nu \in I(\Lambda') \quad \text{si} \quad (\overline{P}_\nu, \overline{\Lambda}') = \sigma_\nu.$$

Il est évident que le mode E_0 de définition de la grandeur θ' de l'étape dans la direction $\overline{l}'$ se ramène au calcul de θ maximal tel que le vecteur $\overline{\Lambda}' + \theta\overline{l}'$ constitue encore le programme du problème (3.8)-(3.10). Si $S(M, \Lambda') < 0$ et $\theta' = \infty$, le problème (3.1)-(3.4),

comme nous le savons, est irréalisable. Si par contre $\theta' < \infty$, on inclut dans l'ensemble $I(\Lambda'')$ l'indice $\nu' \notin I(\Lambda')$ du vecteur $\bar{P}_{\nu'}$ tel que

$$(\bar{P}_{\nu'}, \bar{l}') < 0 \tag{7.4}$$

$$(\bar{\Lambda}'' = \bar{\Lambda}' + \theta'\bar{l}'; \quad (\bar{P}_{\nu'}, \bar{\Lambda}'') = \sigma_{\nu'}).$$

La relation (7.3) conduit à $\nu_\alpha \in I(\Lambda'')$ avec $\alpha = 1, 2, \ldots, t$. Par conséquent, la base optimale du problème $\tilde{\Gamma}(M, \Lambda')$ est la base du programme du problème $\tilde{\Gamma}(M, \Lambda'')$.

L'inégalité (7.4) signifie que l'introduction du vecteur $\bar{P}_{\nu'}$ dans la base (7.2) augmente la fonction économique du problème $\tilde{\Gamma}(M, \Lambda'')$ si seulement le programme admettant cette base est non dégénéré. Or, cette dernière condition résulte de l'hypothèse faite au début de la démonstration. Par conséquent,

$$S(M, \Lambda') < S(M, \Lambda''). \tag{7.5}$$

L'inégalité (7.5) entraîne que lors de la résolution du problème avec emploi du procédé E_0 les directions possibles choisies dans les grandes itérations différentes correspondent à des bases différentes composées de vecteurs $\bar{P}_\nu$ et $-\bar{D}_i$.

Le nombre de telles bases étant fini, on tire la conclusion que quelques grandes itérations suffisent pour obtenir le point Λ^* tel que soit $S(M, \Lambda^*) = 0$, soit $S(M, \Lambda^*) < 0$, $\theta^* = \infty$. Dans le premier cas, le programme optimal du problème $\tilde{\Gamma}(M, \Lambda^*)$ est solution du problème (3.1)-(3.4). La deuxième éventualité indique que les conditions (3.2)-(3.4) sont incompatibles. Le théorème est ainsi démontré.

7.2. Le caractère fini des méthodes de programmation par blocs utilisant le procédé E_0 se déduit de l'inégalité (7.5). Or, si lors du passage du point $\bar{\Lambda}'$ au point $\bar{\Lambda}''$ on applique le procédé $E_{\min}$, d'une façon générale, l'inégalité (7.5) cesse d'être respectée. C'est que dans ce cas la solution du problème $\tilde{\Gamma}(M, \Lambda')$ n'est nullement forcée d'être un programme du problème $\tilde{\Gamma}(M, \Lambda'')$. Aussi la convergence de la procédure de minimisation de la fonction $f(\Lambda)$ qui recourt au procédé $E_{\min}$ doit-elle être explorée à partir de considérations qui diffèrent de celles qui précèdent.

Les problèmes $\Gamma(M, \Lambda)$ et $\tilde{\Gamma}(M, \Lambda)$ dont la résolution assure le choix de la direction possible l issue du point Λ, sont définis en fixant l'ensemble G_Λ qui réunit les programmes optimaux du X_Λ-problème

$$(C - \Lambda A^{(0)}, X) \to \max$$

sous les conditions (3.3), (3.4).

La formulation analytique de l'ensemble G_Λ est donnée par les conditions (3.3), (3.4) et la restriction

$$x_j = 0 \quad \text{si} \quad \Delta_j(\Lambda) > 0, \tag{7.6}$$

où les $\Delta_j(\Lambda)$ sont les critères de sélection des vecteurs des conditions du X_Λ-problème par rapport à une certaine base optimale de ce problème.

Soit $\varepsilon > 0$. Définissons l'ensemble convexe $G_\Lambda(\varepsilon)$ à l'aide des conditions (3.3), (3.4) et de la restriction

$$x_j = 0 \quad \text{si} \quad \Delta_j(\Lambda) > \varepsilon. \tag{7.7}$$

Notons que pratiquement la tentative de prélever à l'aide de la condition (7.6) l'ensemble G_Λ conduit en fait, $\varepsilon > 0$ étant suffisamment petit, à satisfaire aux restrictions (7.7). Dans les calculs réels l'ensemble $G_\Lambda(\varepsilon)$ figure au lieu de G_Λ. C'est pourquoi, lors de la détermination de la direction possible, il est naturel de remplacer G_Λ par $G_\Lambda(\varepsilon)$. Convenons donc de choisir comme direction possible l' issue du point Λ' un vecteur tel que

$$\left.\begin{aligned} &\varphi_1^{(\varepsilon)}(l') = \min_{l \in M} \varphi_1^{(\varepsilon)}(l), \\ \text{où} \quad & \\ &\varphi_1^{(\varepsilon)}(l) = \sup_{X \in G_{\Lambda'}(\varepsilon)} (B^{(0)} - A^{(0)}X, l). \end{aligned}\right\} \tag{7.8}$$

Désignons le problème (7.8) par $\Gamma_\varepsilon(M, \Lambda')$. Son dual $\widetilde{\Gamma}_\varepsilon(M, \Lambda')$ consiste à maximiser la fonction

$$\varphi_2(X) \tag{7.9}$$

sous les conditions

$$X \in G_{\Lambda'}(\varepsilon), \tag{7.10}$$

où $\varphi_2(X)$ est définie par la relation (5.18).

Il est évident que

$$v_0(\varepsilon) = \varphi_1^{(\varepsilon)}(l') = \varphi_2(X'_\varepsilon) \leqslant 0$$

[X'_ε est solution du problème (7.9), (7.10)].

En reprenant les raisonnements correspondants du paragraphe 5.2, il vient

$$A^{(0)}X'_\varepsilon = B^{(0)}, \tag{7.11}$$

si $v_0(\varepsilon) = 0$.

Nous laissons au lecteur le soin de démontrer (cf. exercice 16) que la justification de (7.11) donne lieu à l'inégalité

$$(C, X'_\varepsilon) \geqslant (C, X^*) - \alpha \cdot \varepsilon,$$

où X^* est la solution du problème (3.1)-(3.4);

α, la constante qui ne dépend pas de $\varepsilon > 0$ ni de Λ'.

Ainsi, lorsque le choix de la direction possible fait apparaître que $v_0(\varepsilon) = 0$, on construit simultanément un programme X'_ε

du problème (3.1)-(3.4) qui, avec un $\varepsilon > 0$ suffisamment petit, diffère peu du programme optimal. L'écart entre le programme X'_ε et le programme optimal est évalué à l'aide de l'inégalité évidente

$$(C, X^*) - (C, X'_\varepsilon) \leqslant f(\Lambda') - (C, X'_\varepsilon). \tag{7.12}$$

Mais si

$$v_0(\varepsilon) = \varphi_1^{(\varepsilon)}(l') < 0,$$

l' est alors une direction possible, puisque

$$\varphi_1(l') = \sup_{X \in G_{\Lambda'}} (B^{(0)} - A^{(0)}X, l') \leqslant$$

$$\leqslant \sup_{X \in G_{\Lambda'}(\varepsilon)} (B^{(0)} - A^{(0)}X, l') = \varphi_1^{(\varepsilon)}(l') < 0.$$

7.3. Etablissons la convergence du processus d'obtention d'une solution approchée du problème (3.1)-(3.4), fondé sur le procédé ci-dessus décrit de sélection d'une direction possible.

Théorème 7.2. *Supposons que la direction possible l' soit choisie à partir des relations* (7.8) *et que la quantité θ' soit calculée à l'aide du procédé $E_{\min}$. Si la fonction $f(\Lambda)$ possède un minimum fini, on construit, après un nombre fini de grandes itérations, un programme X_ε du problème* (3.1)-(3.4) *et un point Λ associés par la condition*

$$X_\varepsilon \in G_\Lambda(\varepsilon). \tag{7.13}$$

Si la fonction $f(\Lambda)$ n'est pas bornée inférieurement on établit ce fait à l'une des itérations, ou bien on construit une suite de points $\Lambda^{(k)}$ telle que

$$\lim_{k \to \infty} f(\Lambda^{(k)}) = -\infty. \tag{7.14}$$

La démonstration du théorème 7.2 s'appuie sur une proposition auxiliaire.

Lemme 7.1. *Quel que soit $\varepsilon > 0$, il existe un $\delta > 0$, indépendant du point Λ', tel que avec $|\Lambda - \Lambda'| \leqslant \delta$ et $f(\Lambda') < \infty$*

$$f(\Lambda) - (B^{(0)}, \Lambda) = \sup_{X \in G_{\Lambda'}(\varepsilon)} (C_\Lambda, X). \tag{7.15}$$

Démonstration. Soit Λ' un point arbitraire de l'ensemble R. Classons les vecteurs X_ν, $\nu = 1, 2, \ldots, N$, en deux groupes et rangeons dans le premier ceux d'entre eux qui vérifient la condition (7.7) avec $\Lambda = \Lambda'$. Désignons par $I(\Lambda')$ l'ensemble des indices ν de ces vecteurs. Les vecteurs restants sont rangés dans le deuxième groupe. Rappelons que les vecteurs X_ν sont les sommets ou les vecteurs directeurs unités des arêtes infinies de l'ensemble convexe G défini par les conditions (3.3), (3.4).

Soit $\alpha_\nu > 0$ la valeur minimale d'une composante du vecteur X_ν ($\nu \notin I(\Lambda')$) telle que $x_j^{(\nu)} > 0$ bien que $\Delta_j(\Lambda') > \varepsilon$;

$$\alpha = \min_{\nu \in I(\Lambda')} \alpha_\nu.$$

Considérons le vecteur arbitraire X_ν du deuxième groupe ($\nu \notin I(\Lambda')$). Si X_ν est un sommet, il est clair que

$$(C_{\Lambda'}, X_\nu) = (C_{\Lambda'}, X_{\nu*}) - \sum_{j=1}^{n} \Delta_j(\Lambda') x_j^{(\nu)} \leqslant (C_{\Lambda'}, X_{\nu*}) - \alpha \cdot \varepsilon, \quad (7.16)$$

où $X_{\nu*}$ est le programme optimal du $X_{\Lambda'}$-problème.

Si X_ν est le vecteur directeur d'une arête infinie,

$$X_{\nu*} + \theta X_\nu \in G \quad \text{pour tout} \quad \theta \geqslant 0.$$

Par conséquent,

$$(C_{\Lambda'}, X_\nu) = (C_{\Lambda'}, X_{\nu*} + X_\nu) - (C_{\Lambda'}, X_{\nu*}) \leqslant -\alpha \cdot \varepsilon. \quad (7.17)$$

Choisissons un $\delta > 0$ tel que avec $|l| \leqslant \delta$

$$\max_{1 \leqslant \nu \leqslant N} |(lA^{(0)}, X_\nu)| < \frac{1}{2} \alpha \cdot \varepsilon. \quad (7.18)$$

Considérons un point quelconque Λ tel que

$$|\Lambda - \Lambda'| = |l| \leqslant \delta.$$

Si

$$\sup_{X \in G_{\Lambda'}(\varepsilon)} (C_\Lambda, X) = \infty,$$

il est évident que l'égalité (7.15) est vérifiée.

Soit

$$\sup_{X \in G_{\Lambda'}(\varepsilon)} (C_\Lambda, X) = (C_\Lambda, X_\nu) < \infty, \quad \nu \in I(\Lambda'). \quad (7.19)$$

Pour un vecteur directeur arbitraire X_ν d'une arête infinie de l'ensemble G, on a

$$(C_\Lambda, X_\nu) \leqslant -\frac{1}{2} \alpha\varepsilon \leqslant 0 \text{ avec } \nu \notin I(\Lambda')$$

[en vertu de (7.17) et (7.18)];

$$(C_\Lambda, X_\nu) \leqslant 0 \quad \text{avec} \quad \nu \in I(\Lambda')$$

[en vertu de (7.19)].

Par conséquent

$$\sup_{X \in G} (C_\Lambda, X) = (C_\Lambda, X_{\nu'}) < \infty,$$

où $X_{\nu'}$ est un certain sommet de G. Montrons que $\nu' \in I(\Lambda')$. En effet,

$$(C_\Lambda, X_{\nu'}) \geqslant (C_\Lambda, X_{\nu*}),$$

ou, d'après (7.18),

$$(C_\Lambda, X_{\nu'}) > (C_{\Lambda'}, X_{\nu*}) - \frac{1}{2}\alpha\varepsilon. \tag{7.20}$$

D'autre part, pour tout $\nu \notin I(\Lambda')$, les relations (7.16) et (7.18) entraînent l'inégalité

$$(C_\Lambda, X_\nu) \leqslant (C_{\Lambda'}, X_{\nu*}) - \frac{1}{2}\alpha\varepsilon. \tag{7.21}$$

La comparaison de (7.20) et de (7.21) nous prouve que $X_{\nu'}$ est un sommet de l'ensemble $G_{\Lambda'}(\varepsilon)$. C'est pourquoi

$$\sup_{X \in G} (C_\Lambda, X) = \sup_{X \in G_{\Lambda'}(\varepsilon)} (C_\Lambda, X).$$

Le lemme est ainsi démontré.

7.4. Démonstration du théorème 7.2. 1. De même que le problème $\widetilde{\Gamma}(M, \Lambda')$ [cf. (5.22)-(5.24)], le problème $\widetilde{\Gamma}_\varepsilon(M, \Lambda')$ consiste à maximiser la forme linéaire

$$-\sum_{i=1}^{r} d_i\mu_i$$

sous les conditions

$$\sum_{\nu \in I(\Lambda', \varepsilon)} z_\nu \overline{P}_\nu - \sum_{i=1}^{r} \mu_i \overline{D}_i = \overline{B}^{(0)},$$
$$z_\nu \geqslant 0, \quad \mu_i \geqslant 0,$$

où $I(\Lambda', \varepsilon)$ est l'ensemble des numéros ν des vecteurs X_ν qui participent à la formation de l'ensemble $G_{\Lambda'}(\varepsilon)$.

Le nombre N des vecteurs $\overline{P}_\nu$ étant fini, le nombre total de bases des problèmes $\widetilde{\Gamma}_\varepsilon(M, \Lambda')$ associées à tous les points Λ' possibles ne dépasse pas C_{N+r}^{m+1}. Mais, on sait que la valeur minimale d'une forme linéaire s'obtient sur l'un des programmes de base du problème. Il existe donc un nombre $\gamma > 0$, indépendant du point Λ', tel que la valeur extrémale de $v_0(\varepsilon)$ du problème $\widetilde{\Gamma}_\varepsilon(M, \Lambda')$ est nulle, soit

$$v_0(\varepsilon) = -\min\left(\sum_{i=1}^{r} d_i\mu_i\right) < -\gamma. \tag{7.22}$$

2. Soit l' la direction choisie au terme de la résolution des problèmes $\Gamma_\varepsilon(M, \Lambda')$ et $\widetilde{\Gamma}_\varepsilon(M, \Lambda')$ et, de plus,

$$v_0(\varepsilon) = \varphi_1^{(\varepsilon)}(l') = \max_{X \in G_{\Lambda'}(\varepsilon)} (-l'A^{(0)}, X) + (B^{(0)}, l') < 0. \tag{7.23}$$

On tire de (7.22) que lorsque (7.23) est observée

$$v_0(\varepsilon) = \max_{X \in G_{\Lambda'}(\varepsilon)} (B^{(0)} - A^{(0)}X,\ l') < -\gamma. \tag{7.24}$$

Choisissons un $\theta > 0$ tel que

$$\theta \mid l' \mid = \delta, \tag{7.25}$$

où δ est le nombre qui figure dans le lemme 7.1.

En utilisant le lemme 7.1 et l'inégalité (7.24), on a

$$\begin{aligned} f(\Lambda' + \theta l') - f(\Lambda') &= \sup_{X \in G} (C - (\Lambda' + \theta l') A^{(0)},\ X) - \\ &- \sup_{X \in G} (C - \Lambda' A^{(0)},\ X) + \theta (B^{(0)},\ l') = \\ &= \sup_{X \in G_{\Lambda'}(\varepsilon)} (C - (\Lambda' + \theta l') A^{(0)},\ X) - \\ &- \sup_{X \in G_{\Lambda'}(\varepsilon)} (C - \Lambda' A^{(0)},\ X) + \theta (B^{(0)},\ l') \leqslant \\ &\leqslant \theta \left[\sup_{X \in G_{\Lambda'}(\varepsilon)} (-l' A^{(0)},\ X) + (B^{(0)},\ l') \right] = \theta v_0(\varepsilon) < -\theta\gamma. \end{aligned} \tag{7.26}$$

Pour un ensemble M borné, la longueur du vecteur l' est limitée par un nombre q indépendant du point Λ'. Cette conclusion reste valable dans le cas général, si seulement on choisit l' parmi les programmes de base du problème linéaire équivalent à $\Gamma_\varepsilon(M, \Lambda')$, ce qui est naturellement toujours possible. De là et de l'égalité (7.25) on tire le critère pour θ :

$$\theta \geqslant \frac{\delta}{q}.$$

Par suite, l'inégalité (7.26) conduit à la relation

$$f(\Lambda' + \theta l') \leqslant f(\Lambda') - \frac{\delta}{q}\gamma.$$

Lorsqu'on utilise le procédé $E_{\min}$ la quantité θ' est choisie par minimisation de la fonction $f(\Lambda' + \theta l')$ par rapport à θ, on a pour $\theta' < \infty$

$$f(\Lambda'') = f(\Lambda' + \theta' l') \leqslant f(\Lambda' + \theta l') \leqslant f(\Lambda') - \frac{\delta}{q}\gamma. \tag{7.27}$$

3. Ainsi, lorsque $v_0(\varepsilon) < 0$, $\theta' < \infty$, au terme d'une grande itération la valeur de la fonction $f(\Lambda)$ décroît au moins de la valeur du nombre fixe $\frac{\delta}{q}\gamma > 0$. Par conséquent, lorsque les conditions (3.2)-(3.4) sont compatibles et $f(\Lambda)$ possède un minimum fini, quelques grandes itérations permettent d'obtenir le point Λ tel que $v_0(\varepsilon) = 0$ et de construire en même temps le programme X_ε

du problème (3.1)-(3.4) associé à Λ par la relation (7.13). Mais si

$$\inf f(\Lambda) = -\infty,$$

et de plus, si à chaque grande itération $\theta' < \infty$, il vient

$$f(\Lambda^{(k)}) \leqslant f(\Lambda') - k\frac{\delta}{q}\gamma,$$

c'est-à-dire

$$\lim_{k\to 0} f(\Lambda^{(k)}) = -\infty,$$

ce qu'il fallait démontrer.

§ 8. Autre méthode de programmation par blocs

8.1. Dans certains cas, outre le point $\Lambda' \in R$ tel que

$$f(\Lambda') < \infty,$$

on connaît également une des bases du Z-problème (3.5)-(3.7).

Il en est ainsi, par exemple, lorsqu'on recourt à la méthode de programmation par blocs exposée au paragraphe précédent.

Si $\varepsilon > 0$ définissant les ensembles $G_\Lambda(\varepsilon)$ n'est pas assez petit, le programme du problème (3.1)-(3.4) obtenu par minimisation de la fonction $f(\Lambda)$ peut différer beaucoup de sa valeur optimale et doit donc être soumis à l'amélioration. Toutefois, nous avons déjà à notre disposition non seulement le point $\Lambda' \in R$, mais encore la base du programme du Z-problème associé au programme établi du problème (3.1)-(3.4). Dans la méthode de programmation par blocs donnée ci-dessous, l'optimum est recherché par minimisation de la fonction $f(\Lambda)$ et par amélioration simultanée du programme disponible. Voici l'interprétation géométrique de cette méthode. Parmi les hyperplans qui engendrent l'ensemble convexe $\bar{R}$ on choisit ceux qui ont des points communs avec l'ensemble $\bar{R}$ dans un certain voisinage fixé du point $\bar{\Lambda}' = (\Lambda', f(\Lambda'))$, ou bien qui correspondent aux vecteurs de la base disponible du Z-problème.

Les hyperplans choisis engendrent l'ensemble convexe

$$\bar{R}(\Lambda';\ P_X, \varepsilon) \supset \bar{R},$$

où P_X est la base du Z-problème associé au programme du X-problème (3.1)-(3.4);

$\varepsilon \geqslant 0$ définit le voisinage du point $\bar{\Lambda}'$.

On cherche ensuite le point « le plus bas » $(\bar{\Lambda}', f(\bar{\Lambda}'))$ de l'ensemble $\bar{R}(\Lambda', P_X, \varepsilon)$, associé à une base nouvelle $P_{X'}$ du problème (3.5)-(3.7).

Si

$$f(\Lambda') > (C, X),$$

le vecteur $l' = \hat{\Lambda}' - \Lambda'$ détermine la direction dans laquelle décroît la fonction $f(\Lambda)$. En se déplaçant dans cette direction, on obtient un nouveau point Λ''. Le passage du point Λ' et de la base P_X à un nouveau couple Λ'', $P_{X'}$ constitue une grande itération de la méthode considérée.

8.2. Exposons cette méthode plus en détail. Supposons donc que nous connaissions le point Λ' et la base P_X du programme de base du problème (3.5)-(3.7) associé au programme X du problème initial. Le point Λ' définit l'ensemble $G_{\Lambda'}(\varepsilon)$, qui, comme nous le savons, est donné par les conditions (3.3), (3.4) et la contrainte supplémentaire

$$x_j = 0 \quad \text{si} \quad \Delta_j(\Lambda') > \varepsilon,$$

où les $\Delta_j(\Lambda')$ sont les critères de sélection des vecteurs des conditions du $X_{\Lambda'}$-problème par rapport à l'une de ses bases optimales.

Soit $\overline{Y} = (Y, y_{m+1}) = (y_1, y_2, \ldots, y_m, y_{m+1})$ le vecteur des critères de sélection des conditions du Z-problème par rapport à la base P_X.

Composons le $Y(\Lambda', \varepsilon)$-problème qui consiste à maximiser la forme linéaire

$$(C - YA^{(0)}, X)$$

sous les conditions

$$X \in G_{\Lambda'}(\varepsilon).$$

Conformément à la description de la méthode de décomposition, on détermine, au terme de la résolution du $Y(\Lambda', \varepsilon)$-problème, le vecteur $\overline{P}_{\nu*}$ à inclure dans la base P_X (X_ν est le sommet ou le vecteur directeur d'une arête infinie de l'ensemble $G_{\Lambda'}(\varepsilon)$), ou bien on met en évidence l'impossibilité d'une augmentation ultérieure de la forme linéaire (3.1) qui serait due à l'emploi des vecteurs associés à l'ensemble $G_{\Lambda'}(\varepsilon)$.

Cette dernière circonstance a lieu quand il faut justifier l'égalité

$$(C - YA^{(0)}, X^*) = y_{m+1}, \tag{8.1}$$

où X^* est solution du $Y(\Lambda', \varepsilon)$-problème.

Dans le premier cas, la base P_X se transforme et l'on compose un nouveau $Y'(\Lambda', \varepsilon)$-problème pour lequel on effectue les mêmes opérations que pour le précédent. En quelques étapes nous aboutissons au $Y^{(k)}(\Lambda', \varepsilon)$-problème qui vérifie l'égalité (8.1). Ce problème est engendré par le vecteur $\overline{Y}^{(k)} = (Y^{(k)}, y_{m+1}^{(k)})$.

Posons $\hat{\Lambda}' = Y^{(k)}$, et soit X' le programme du problème (3.1)-(3.4) associé à la dernière base du Z-problème.

Il est clair que

$$(C,\ X') = y_{m+1}^{(k)} + (\hat{\Lambda}',\ B^{(0)}). \tag{8.2}$$

L'égalité (8.1) avec $\bar{Y} = \bar{Y}^{(k)}$ et la relation (8.2) entraînent

$$\max_{X \in G_{\Lambda'}(\varepsilon)} (C - \hat{\Lambda}' A^{(0)},\ X) + (\hat{\Lambda}',\ B^{(0)}) = (C,\ X'). \tag{8.3}$$

Si

$$f(\Lambda') = (C,\ X'),$$

on sait alors que $f(\Lambda')$ est le minimum cherché de la fonction $f(\Lambda)$, et X' la solution du problème (3.1)-(3.4).

Mais si

$$f(\Lambda') > (C,\ X'), \tag{8.4}$$

on vérifie aisément que le vecteur

$$l' = \hat{\Lambda}' - \Lambda'$$

définit la direction dans laquelle décroît la fonction $f(\Lambda)$.

En effet,

$$\left.\frac{df(\Lambda)}{dl'}\right|_{\Lambda=\Lambda'} = \sup_{X \in G_{\Lambda'}} (B^{(0)} - A^{(0)}X,\ l') =$$

$$= \sup_{X \in G_{\Lambda'}} [(C - \hat{\Lambda}' A^{(0)},\ X) - (C - \Lambda' A^{(0)},\ X)] +$$

$$+ (\hat{\Lambda}',\ B^{(0)}) - (\Lambda',\ B^{(0)}) = \sup_{X \in G_{\Lambda'}} (C - \hat{\Lambda}' A^{(0)},\ X) +$$

$$+ (\hat{\Lambda}',\ B^{(0)}) - f(\Lambda').$$

Tenant compte ensuite de l'égalité (8.3) et de l'inclusion

$$G_{\Lambda'} \subset G_{\Lambda'}(\varepsilon),$$

on a

$$\left.\frac{df(\Lambda)}{dl'}\right|_{\Lambda=\Lambda'} \leqslant (C,\ X') - f(\Lambda') < 0.$$

L'existence de l'inégalité (8.4) permet ainsi de diminuer la valeur de la fonction $f(\Lambda)$ en se déplaçant dans la direction $l' = \hat{\Lambda}' - \Lambda'$.

Ce déplacement s'effectue d'habitude à l'aide d'une méthode de programmation paramétrique, en partant du programme optimal connu du $X_{\Lambda'}$-problème. Le choix de la longueur de l'étape dans la direction l' peut se faire soit d'après le mode E_0 (se déplacer jusqu'à la première valeur critique positive de la fonction $f(\Lambda' + \theta l')$), soit à l'aide du procédé $E_{\min}$ (se déplacer tant que la fonction $f(\Lambda' + \theta l')$ décroît). Constatons que dans la méthode considérée

la longueur de l'étape est toujours finie, puisque pour tout Λ

$$f(\Lambda) \geqslant (C, X),$$

où X est le programme initial du problème (3.1)-(3.4).

Soit θ' la longueur choisie de l'étape dans la direction l'. Dans ce cas,

$$\Lambda'' = \Lambda' + \theta' l'.$$

On obtient ainsi un nouveau point $\Lambda'' \in R$ et une nouvelle base $P_{X'}$ du problème (3.5)-(3.7) associée au programme X' du problème (3.1)-(3.4). On connaît également la base du programme optimal du $X_{\Lambda''}$-problème qui permet de constituer les contraintes supplémentaires

$$x_j = 0 \quad \text{si} \quad \Delta_j(\Lambda'') > \varepsilon,$$

qui définissent avec les conditions (3.3), (3.4) le polyèdre convexe $G_{\Lambda''}(\varepsilon)$. Les nouvelles données servent de base à l'exécution de la grande itération suivante.

La démonstration de la finitude de la méthode ainsi décrite n'exige pas de grands efforts.

Si la longueur d'une étape est définie par le mode $E_{\min}$ et le couple de points X'', Λ'' vérifie la condition (8.4), alors $P_{X'} \neq P_{X''}$, car dans le cas contraire on aurait $\hat{\Lambda}'' = \hat{\Lambda}'$, et la fonction $f(\Lambda' + \theta(\hat{\Lambda}' - \Lambda'))$ ne pourrait atteindre le minimum avec $\theta = \theta'$ ($\Lambda'' = \Lambda' + \theta'(\hat{\Lambda}' - \Lambda')$). Des raisonnements analogues montrent que lorsqu'on utilise le procédé E_0, la base $P_{X'}$ ne peut se conserver que dans un nombre fini de grandes itérations. Dans le cas non dégénéré, chaque changement de base du Z-problème s'accompagne d'un accroissement de la valeur de sa forme linéaire.

C'est pourquoi toutes les bases obtenues sont différentes. Lorsqu'on utilise les règles usuelles qui rendent impossible le cyclage, cette même conclusion reste valide pour le cas général. Par conséquent, la procédure ne peut durer indéfiniment.

8.3. A titre de conclusion à ce paragraphe, notons une fois encore que certaines méthodes de programmation par blocs examinées dans ce chapitre peuvent être considérées comme des généralisations des méthodes finies bien connues de la programmation linéaire.

La méthode de décomposition de Dantzig et Wolfe est fondée sur la méthode primale du simplexe. La méthode décrite au § 4 est l'analogue de la méthode duale du simplexe. Comme on l'a signalé au paragraphe 5.8, le cas particulier de la classe des méthodes de programmation par blocs associées à la minimisation de la fonction $f(\Lambda)$ est une modification prévue pour le problème décomposé en blocs de la méthode primale-duale. Enfin, l'algorithme exposé dans le présent paragraphe se fonde sur la méthode des encadrements (cf. [87], chapitre 10, § 4 ou [21]).

Dans chacune des quatre méthodes mentionnées de programmation par blocs on utilise pour l'analyse du Z-problème la méthode finie respective de programmation linéaire.

Une vision d'ensemble de la formation des méthodes de programmation par blocs examinée de près aux §§ 3 à 8 conduit à une série d'algorithmes qui ne coïncident avec aucune des quatre méthodes exposées, bien qu'ils soient plus proches des deux dernières que de toute autre méthode.

§ 9. Application de la méthode de décomposition au problème de transport et à ses variantes

9.1. Les méthodes de programmation par blocs s'avèrent très utiles lors de la résolution de nombreux problèmes spéciaux de programmation linéaire.

Les méthodes spéciales de résolution des problèmes comme le problème de transport, le problème de distribution, etc., simplifient nettement les calculs par rapport aux méthodes générales de programmation linéaire. Toutefois, les méthodes spéciales sont très intimement liées à la structure de la matrice des conditions du problème. Dans la pratique on se trouve souvent devant la nécessité de faire appel, en plus des conditions de type spécial associées, entre autres, au problème de transport ou de distribution, à un petit nombre de contraintes linéaires de type général. L'application des algorithmes spéciaux dans de tels cas se complique considérablement. Les méthodes de programmation par blocs (et notamment le principe de décomposition), utiles par elles-mêmes lors de la résolution de nombreux problèmes de structure spéciale, se trouvent être particulièrement efficaces lors de la résolution de problèmes spéciaux « gâchés » par des conditions supplémentaires de type général (cf. [23]).

Dans [14] on insiste sur le fait que l'application de la méthode de décomposition à la résolution d'un problème de transport dans lequel le nombre de points de consommation dépasse nettement celui de points de production (ou, inversement, $n \ll m$) est liée à des calculs moins complexes que les autres méthodes spéciales connues de résolution des problèmes de transport rationnel. Il est donc d'autant plus avantageux d'utiliser la méthode de décomposition pour résoudre les problèmes de transport complétés par des conditions de type général.

Le présent paragraphe a donc pour objet l'examen des possibilités qu'offre la programmation par blocs dans le domaine des problèmes de transport et de distribution et de leurs variantes.

9.2. Le problème de transport classique consiste à calculer les paramètres x_{ij} $(i = 1, 2, \ldots, m;\ j = 1, 2, \ldots, n)$ tels qu'ils

rendent minimale la forme linéaire

$$L = \sum_{j=1}^{n} \sum_{i=1}^{m} c_{ij} x_{ij} \tag{9.1}$$

sous les conditions

$$\sum_{j=1}^{n} x_{ij} = a_i, \quad i = 1, 2, \ldots, m; \tag{9.2}$$

$$\sum_{i=1}^{m} x_{ij} = b_j, \quad j = 1, 2, \ldots, n; \tag{9.3}$$

$$x_{ij} \geqslant 0, \quad i = 1, 2, \ldots, m; \quad j = 1, 2, \ldots, n. \tag{9.4}$$

Considérons le problème de transport comme un problème de programmation par blocs dans lequel les conditions d'égalité sont divisées en deux blocs définis par les égalités (9.2) et (9.3) respectivement. Désignons par $A^{(0)}$ la matrice qui vérifie les contraintes (9.2) et par $A^{(1)}$ la matrice des conditions (9.3).

Appelons X'-problème celui qui consiste à minimiser la forme linéaire (9.1) sous les conditions (9.3), (9.4). Il est aisé de comprendre que l'on peut obtenir tous les programmes de base du X'-problème si l'on pose pour tout j un des x_{ij} égal à b_j et les autres égaux à zéro. Le nombre total de programmes de base du X'-problème s'écrit $N = m^n$. Désignons par $X_\nu = \{x_{ij}^{(\nu)}\}$, $\nu = 1, 2, \ldots, N$, les programmes de base du X'-problème.

Les conditions (9.3), (9.4) du X'-problème définissent un polyèdre convexe. C'est pourquoi un programme quelconque de ce problème peut être mis sous la forme

$$X = \sum_{\nu=1}^{N} X_\nu z_\nu, \tag{9.5}$$

où

$$\sum_{\nu=1}^{N} z_\nu = 1, \tag{9.6}$$

$$z_\nu \geqslant 0, \quad \nu = 1, 2, \ldots, N. \tag{9.7}$$

Considérant le problème de transport comme un X-problème, écrivons le Z-problème qui lui correspond, et qui consiste à minimiser la forme linéaire

$$\sum_{\nu=1}^{N} c_\nu z_\nu \tag{9.8}$$

sous les conditions

$$\sum_{\nu=1}^{N} P_\nu z_\nu = a, \tag{9.9}$$

$$z_\nu \geqslant 0, \quad \nu = 1, 2, \ldots, N. \tag{9.10}$$

Ici, le vecteur $a = (a_1, a_2, \ldots, a_m)^T$, les coefficients c_ν de la forme linéaire (9.8) se calculent d'après la formule (1.11) et les vecteurs P_ν sont déterminés par la relation (1.12). Tenant compte de la forme de la matrice $A^{(0)}$ des conditions (9.2), on obtient les expressions suivantes pour les coefficients c_ν et les composantes $p_\nu^{(i)}$ des vecteurs des conditions P_ν:

$$c_\nu = \sum_{j=1}^{n} \sum_{i=1}^{m} c_{ij} x_{ij}^{(\nu)}, \qquad \nu = 1, 2, \ldots, N, \tag{9.11}$$

$$p_\nu^{(i)} = \sum_{j=1}^{n} x_{ij}^{(\nu)}, \quad i = 1, 2, \ldots, m; \quad \nu = 1, 2, \ldots N. \tag{9.12}$$

(Chaque composante $p_\nu^{(i)}$ du vecteur P_ν est égale à la somme des éléments de la ligne correspondante (i-ème) de la matrice x_{ij}^{ν}. Cela signifie que $p_\nu^{(i)}$ est soit nul, soit égal à la somme de certains des nombres b_j.)

D'après les règles de construction du Z-problème [cf. paragraphe 1.1., problème (1.13)-(1.16)], il doit encore y avoir parmi ses contraintes une condition du type (1.15). Pourtant on voit sans peine que cette contrainte se déduit des conditions (9.9).

En effet, en additionnant les conditions (9.9) suivant les lignes, on obtient

$$\sum_{i=1}^{m} \sum_{\nu=1}^{N} p_\nu^{(i)} z_\nu = \sum_{i=1}^{m} a_i,$$

ou, en vertu de (9.12),

$$\sum_{\nu=1}^{N} \sum_{i=1}^{m} \sum_{j=1}^{n} x_{ij}^{(\nu)} z_\nu = \sum_{i=1}^{m} a_i.$$

$\| x_{ij}^{(\nu)} \|$ étant le programme du X'-problème, il vient

$$\sum_{\nu=1}^{N} \left(\sum_{i=1}^{m} a_i \right) z_\nu = \sum_{i=1}^{m} a_i.$$

Ainsi, la contrainte

$$\sum_{\nu=1}^{N} z_\nu = 1$$

se déduit des égalités (9.9).

Les raisonnements effectués ramènent la résolution du problème de transport classique à la détermination du programme optimal du problème de type général suivant.

Minimiser la forme linéaire

$$\sum_{\nu=1}^{N} c_\nu z_\nu \tag{9.13}$$

sous les conditions

$$\sum_{\nu=1}^{N} p_\nu^{(i)} z_\nu = a_i, \qquad i = 1, 2, \ldots, m, \tag{9.14}$$

$$z_\nu \geqslant 0, \quad \nu = 1, 2, \ldots, N. \tag{9.15}$$

Au cours de la résolution du Z-problème, on calcule à chaque itération le vecteur Λ de dimension m des critères de sélection des conditions par rapport à la base courante. Les composantes λ_i du vecteur Λ vérifient le système d'équations

$$\sum_{\mu=1}^{m} p_{\nu_i}^{(\mu)} \lambda_\mu = c_{\nu_i}, \qquad i = 1, 2, \ldots, m, \tag{9.16}$$

où les ν_i ($i = 1, \ldots, m$) sont les numéros des vecteurs de la base du Z-problème.

En se servant du vecteur Λ, on construit le X_Λ-problème. D'après les règles générales de la méthode de décomposition, la forme linéaire du X_Λ-problème s'écrit

$$L_\Lambda(X) = \sum_{j=1}^{n} \sum_{i=1}^{m} (c_{ij} - \lambda_i)\, x_{ij}. \tag{9.17}$$

Le domaine de définition du X_Λ-problème coïncide avec celui du X'-problème et est délimité par les contraintes (9.3), (9.4). Par suite, le programme optimal du X_Λ-problème, en tant que programme de base du X'-problème, est un vecteur $X_{\nu*} = \{x_{ij}^{(\nu*)}\}$ de dimension $m \cdot n$ dans lequel pour chaque j il n'y a qu'une composante différente de zéro (et égale à b_j). Le numéro s de la composante non nulle se calcule pour chacun des j à partir de la relation

$$c_{sj} - \lambda_s = \min_i (c_{ij} - \lambda_i).$$

Si le minimum est atteint sur plusieurs indices, on peut choisir pour s n'importe lequel d'entre eux.

Au programme de base $X_{\nu*}$ correspond le vecteur $P_{\nu*}$ des conditions du Z-problème. Le critère du vecteur $P_{\nu*}$ par rapport à la base courante est de grandeur égale et de signe opposé à la valeur de la forme linéaire du X_Λ-problème sur son programme optimal $X_{\nu*}$:

$$\Delta_{\nu*} = -L_\Lambda(X_{\nu*}) = -\sum_{j=1}^{n} \sum_{i=1}^{m} (c_{ij} - \lambda_i)\, x_{ij}^{(\nu*)}.$$

L'égalité $\Delta_{\nu*} = 0$ témoigne de ce que la base étudiée du Z-problème correspond à son programme optimal.

Le programme optimal du problème de transport se calcule dans ce cas d'après la formule

$$X^* = \sum_{\nu=1}^{N} X_\nu z_\nu^*, \tag{9.18}$$

où $\{z_\nu^*\}$ est la solution du Z-problème.

L'inégalité $\Delta_{\nu*} > 0$ signifie qu'il faut poursuivre la procédure de résolution. Il faut introduire dans la base du Z-problème le vecteur $P_k = P_{\nu*}$ aux composantes

$$p_k^{(i)} = \sum_{j=1}^{n} x_{ij}^{(\nu*)}.$$

Le vecteur à exclure de la base est déterminé selon les règles générales de la méthode primale du simplexe.

Un nombre fini d'étapes nous amène au X_Λ-problème tel que la valeur de la forme linéaire sur le programme optimal est égale à zéro. Le programme de base correspondant du Z-problème définit selon la formule (9.18) la solution du problème de transport initial.

9.3. Notons certaines particularités du schéma de calcul associé à la résolution du problème de transport par la méthode de décomposition.

Pour calculer les composantes $p_\nu^{(i)}$ des vecteurs des conditions et les coefficients c_ν de la forme linéaire du Z-problème, il est commode d'utiliser le tableau condensé des X_ν qui caractérise les programmes de base étudiés des X_Λ-problèmes.

Le tableau des X_ν est utile également pour le calcul du programme optimal du problème de transport initial.

Pour chaque j, on porte dans la ligne ν la valeur de l'indice $i = i(j)$ de la composante non nulle du programme de base X_ν. Autrement dit, le tableau des X_ν contient les indices s calculés d'après la formule

$$c_{sj} - \lambda_s = \min_i (c_{ij} - \lambda_i).$$

Les composantes $p_\nu^{(i)}$ du vecteur des conditions P_ν se calculent selon la formule

$$p_\nu^{(i)} = \sum_{j=1}^{n} b_{ij}^{(\nu)}, \tag{9.19}$$

où $b_{ij}^{(\nu)}$ est égal à b_j ou à zéro suivant que l'indice i figure à l'intersection de la colonne j et de la ligne ν ou non. Ainsi, la composante $p_\nu^{(i)}$ est nulle si l'indice i ne figure pas dans la ν-ème ligne.

Les coefficients c_ν de la forme linéaire du Z-problème se calculent d'après la relation

$$c_\nu = \sum_{j=1}^{n} b_{i\nu(j),\, j} c_{i\nu(j),\, j} = \sum_{i=1}^{m} \sum_{j=1}^{n} b_{ij}^{(\nu)} c_{ij}. \tag{9.20}$$

Supposons que la base du programme optimal du Z-problème se compose des vecteurs $P_{\nu_1}, P_{\nu_2}, \ldots, P_{\nu_m}$. Cette base est engendrée par les programmes de base $X_{\nu_1}, X_{\nu_2}, \ldots, X_{\nu_m}$ du système (9.3), (9.4). Les composantes du programme optimal du problème de transport se calculent à partir des variables de base z_ν^* de la solution du Z-problème d'après la formule:

$$x_{ij}^* = \sum_{s=1}^{m} b_{ij}^{(\nu_s)} z_{\nu_s}^*. \tag{9.21}$$

Pour achever la description de l'algorithme de résolution du problème de transport par la méthode de décomposition, signalons qu'on peut choisir pour base initiale du Z-problème un système de m vecteurs aux composantes

$$p_{\nu_t}^{(i)} = \begin{cases} \sum_{j=1}^{n} b_j = \sum_{i=1}^{m} a_i = T & \text{avec } t = i, \\ 0 & \text{avec } t \neq i, \end{cases} \tag{9.22}$$
$$t = 1, 2, \ldots, m.$$

La collection considérée de vecteurs des conditions du Z-problème correspond aux programmes de base $X_{\nu_1}, X_{\nu_2}, \ldots, X_{\nu_m}$ du système (9.3), (9.4), dont les composantes se calculent d'après la formule

$$x_{ij}^{(\nu_t)} = \begin{cases} b_j & \text{avec } i = t, \\ 0 & \text{avec } i \neq t. \end{cases}$$

Constatons que plusieurs observations du § 2 sur les variantes possibles de la méthode de décomposition sont notamment applicables aux problèmes de transport et à d'autres problèmes spéciaux de programmation linéaire.

D'autres méthodes d'utilisation de la programmation par blocs, différentes de celle de décomposition (cf. §§ 4, 5, 8), peuvent également être appliquées pour rendre plus rationnelle la résolution du problème de transport. Nous laissons au lecteur le soin de s'en convaincre par lui-même (cf. exercice 20).

L'expérience des calculs atteste que pour $m \ll n$, la résolution d'un problème de transport par les méthodes de programmation par blocs impose des calculs plus simples que la mise en œuvre des méthodes spéciales. Ainsi, dans [14] on constate que la résolution d'un problème de dimension $15 \times 3\,000$ par la méthode de décom-

Tableau des X_ν

ν \ j	1	2	3		...		n
1	$i^{(1)}(1)$	$i^{(1)}(2)$	$i^{(1)}(3)$	...	...	...	$i^{(1)}n$
2	$i^{(2)}(1)$	$i^{(2)}(2)$	$i^{(2)}(3)$	...	...	...	$i^{(2)}(n)$
⋮	...	...	...	...	...	...	...
⋮	...	...	...	...	...	...	...
⋮	...	...	...	...	...	...	...
ν	$i^{(\nu)}(1)$	$i^{(\nu)}(2)$	$i^{(\nu)}(3)$	...	...	...	$i^{(\nu)}(n)$
⋮	...	...	...	...	...	...	...
⋮	...	...	...	...	...	...	...
⋮	...	...	...	...	...	...	...

position a demandé une fois et demie moins de temps que l'utilisation d'un programme compact établi d'après une des méthodes classiques de calcul du programme optimal. La résolution d'un tel problème sur un ordinateur 32K ИБМ-704 a demandé respectivement 56 et 80 minutes.

Il convient de noter que l'algorithme de résolution d'un problème de transport, qui utilise le principe de décomposition, est bien plus simple logiquement que les algorithmes spéciaux connus conçus pour l'analyse de ce problème.

9.4. Illustrons les recommandations des paragraphes précédents par l'exemple d'un problème de transport à matrice des coefficients

$$C=\{c_{ij}\}=\begin{Vmatrix} 1 & 3 & 2 & 1 & 5 \\ 4 & 2 & 3 & 2 & 1 \end{Vmatrix},$$

les volumes de production et de consommation étant respectivement

$$b_1 = 15;\ b_2 = 9;\ b_3 = 12;\ b_4 = 8;\ b_5 = 6;$$
$$a_1 = 31;\ a_2 = 19.$$

On a

$$\Gamma = \sum_{i=1}^{2} a_i = \sum_{j=1}^{5} b_j = 50.$$

La base initiale du Z-problème est définie par la matrice [cf. (9.22)]

$$\| P_1, P_2 \| = \begin{Vmatrix} T & 0 \\ 0 & T \end{Vmatrix} = \begin{Vmatrix} 50 & 0 \\ 0 & 50 \end{Vmatrix} = 50 \begin{Vmatrix} 1 & 0 \\ 0 & 1 \end{Vmatrix}.$$

Les vecteurs de cette base sont associés aux programmes de base X_1 et X_2 du

Tableau 4.12

	N	C_v	B_z	$e_0 = z_{i0}$	e_1	e_2	P_k	θ	N_z
←	1	104	P_1	0,62	0,02		0,7	$\boxed{0,89}$	
	2	136	P_2	0,38		0,02	0,3	1,27	0
—	3	—	—	116,16	2,08	2,72	42,6	—	
→	1	71	P_3	0,89	0,0286		0,42	2,06	
←	2	136	P_2	0,11	—0,0086	0,02	0,58	$\boxed{0,2}$	1
—	3	—	—	78,5	0,86	2,72	17,15	—	
	1	71	P_3	0,8	0,035	—0,015	0,4	2	
→ ←	2	91	P_4	0,2	—0,015	0,035	0,6	$\boxed{0,33}$	2
—	3	—	—	75	1,12	2,12	0	—	
	1	71	P_3	0,667					
→	2	83	P_5	0,333					3
—	3	—	—	13,333				—	

X'-problème:

$$X_1=\begin{Vmatrix} 15 & 9 & 12 & 8 & 6 \\ 0 & 0 & 0 & 0 & 0 \end{Vmatrix}; \quad X_2=\begin{Vmatrix} 0 & 0 & 0 & 0 & 0 \\ 15 & 9 & 12 & 8 & 6 \end{Vmatrix}.$$

On en tire que la première ligne ($\nu = 1$) du tableau des X_ν (tableau 4.13) se compose des unités, et la deuxième ligne ($\nu = 2$), des deux.

Tableau X_ν (4.13)

	ν \ j	1	2	3	4	5
	1	1	1	1	1	1
	2	2	2	2	2	2
*	3	1	2	1	1	2
	4	1	2	2	2	2
*	5	1	2	2	1	2

D'après la formule (9.20), les coefficients de la forme linéaire du Z-problème correspondant aux vecteurs P_1 et P_2 sont égaux respectivement à

$$c_1=\sum_{j=1}^{n} c_{1j}b_j=104; \quad c_2=\sum_{j=1}^{n} c_{2j}b_j=136.$$

Les critères des conditions du Z-problème par rapport à la base initiale se calculent selon des formules du type (9.16)

$$\begin{Vmatrix} 50 & 0 \\ 0 & 50 \end{Vmatrix} \begin{pmatrix} \lambda_1 \\ \lambda_2 \end{pmatrix} = \begin{pmatrix} 104 \\ 136 \end{pmatrix}.$$

D'où

$$\lambda_1 = 2{,}08; \quad \lambda_2 = 2{,}72.$$

La matrice des coeffcients de la forme linéaire du X_Λ-problème [cf. (9.17)] s'écrit dans le cas considéré

$$\| c_{ij}-\lambda_i \| = \begin{Vmatrix} -1{,}08 & 0{,}92 & -0{,}08 & -1{,}08 & 2{,}92 \\ 1{,}28 & -0{,}72 & 0{,}28 & -0{,}72 & -1{,}72 \end{Vmatrix}.$$

Le minimum de la forme linéaire du X_Λ-problème s'obtient sur le programme de base

$$X_3=\begin{Vmatrix} 15 & 0 & 12 & 8 & 0 \\ 0 & 9 & 0 & 0 & 6 \end{Vmatrix}.$$

Ainsi, dans la troisième ligne ($\nu = 3$) du tableau des X_ν, les unités ($i = 1$) sont portées dans les première, troisième et quatrième cases, et les deux dans les deuxième et cinquième cases.

La valeur de la forme linéaire du X_Λ-problème sur le programme X_3 est égale à $L_\Lambda(X_3) = -42{,}6$. Le programme X_3 engendre le vecteur des conditions P_3 au critère $\Delta_3 = -L_\Lambda(X_3) = 42{,}6 > 0$. Le signe positif du critère Δ_3 traduit le fait que la base initiale du Z-problème n'est pas sa base optimale. Il faut ajouter à la base le vecteur P_3 dont les composantes sont d'après la formule (9.19) $p_3^{(1)} = 35$ et $p_3^{(2)} = 15$.

La marche ultérieure de la résolution du problème est donnée par le tableau 4.12 (tableaux principaux de la résolution du Z-problème par le deuxième algorithme primal du simplexe) et le tableau 4.13 (tableau X_ν).

Dans le tableau X_ν, les astérisques marquent les lignes correspondant à la base optimale du Z-problème.

Le programme optimal du problème de transport se calcule selon la formule (9.21):

$$X^* = 0{,}667X_3 + 0{,}333X_5 = \begin{Vmatrix} 15 & 0 & 8 & 8 & 0 \\ 0 & 9 & 4 & 0 & 6 \end{Vmatrix}.$$

9.5. En considérant le problème de distribution comme un problème de programmation par blocs, on aboutit à un algorithme voisin du schéma de résolution du problème de transport décrit dans le paragraphe précédent.

Le problème de distribution (cf. [87], chapitre 3, § 2 et chapitre 15) consiste à rechercher les variables x_{ij} qui permettent de minimiser la forme linéaire

$$L = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij}x_{ij} \tag{9.23}$$

sous les conditions

$$\sum_{j=1}^{n} x_{ij} \leqslant a_i, \quad i = 1, 2, \ldots, m, \tag{9.24}$$

$$\sum_{i=1}^{m} k_{ij}x_{ij} = b_j, \quad j = 1, 2, \ldots, n. \tag{9.25}$$

$$x_{ij} \geqslant 0, \quad i = 1, 2, \ldots, m; \quad j = 1, 2, \ldots, n. \tag{9.26}$$

Supposons tout d'abord que $k_{ij} > 0$ pour tous les i et j. Cela signifie que le polyèdre convexe défini par les conditions (9.25)-(9.26) est borné.

Comme dans le cas du problème de transport, nous rapporterons les conditions (9.24) au bloc à matrice $A^{(0)}$ et le système des contraintes (9.25) au bloc à matrice $A^{(1)}$.

Les coordonnées de tous les sommets du polyèdre défini par les contraintes (9.25)-(9.26) peuvent être obtenues en égalant à zéro, pour tout j, toutes les composantes x_{ij} sauf une. Les coordonnées non nulles des sommets sont égales à $\frac{b_j}{k_{ij}}$.

A un problème de distribution correspond le Z-problème qui consiste à minimiser la forme linéaire

$$\sum_{\nu=1}^{N} c_\nu z_\nu \tag{9.27}$$

sous les conditions

$$\sum_{\nu=1}^{N} p_\nu^{(i)} z_\nu \leqslant a_i, \quad i=1, 2, \ldots, m, \tag{9.28}$$

$$\sum_{\nu=1}^{N} z_\nu = 1, \tag{9.29}$$

$$z_\nu \geqslant 0, \quad \nu = 1, 2, \ldots, N. \tag{9.30}$$

De même que dans le problème de transport, les composantes $p_\nu^{(i)}$ du vecteur P_ν s'écrivent

$$p_\nu^{(i)} = \sum_{j=1}^{n} x_{ij}^{(\nu)},$$

$$c_\nu = \sum_{j=1}^{n} \sum_{i=1}^{m} c_{ij} x_{ij}^{(\nu)}.$$

Dans le Z-problème associé au problème de distribution, la contrainte (9.29) doit être prise en compte comme une condition indépendante. Ceci distingue l'application de la méthode de décomposition au problème de distribution de son utilisation pour la résolution du problème de transport, où la contrainte du type (9.29) se déduit des conditions de résolubilité.

En résolvant le Z-problème (9.27)-(9.30) à l'aide du deuxième algorithme primal du simplexe on obtient à chaque itération le vecteur des critères des conditions

$$\Lambda = (\lambda_1^{(1)}, \lambda_2^{(1)}, \ldots, \lambda_m^{(1)}, \lambda^{(2)}).$$

Le X_Λ-problème correspondant consiste à minimiser la forme linéaire

$$L_\Lambda(X) = \sum_{j=1}^{n} \sum_{i=1}^{m} (c_{ij} - \lambda_i^{(1)}) x_{ij}$$

sous les conditions

$$\sum_{i=1}^{m} k_{ij} x_{ij} = b_j, \quad j=1, 2, \ldots, n,$$

$$x_{ij} \geqslant 0, \quad i = 1, 2, \ldots, m; \quad j = 1, 2, \ldots, n.$$

La solution du X_Λ-problème est donnée par le vecteur $X_{\nu*}$. Pour tout j, une seule des composantes $x_{ij}^{(\nu*)}$ diffère de zéro. L'indice s de cette composante se calcule d'après la relation

$$\frac{(c_{sj} - \lambda_s^{(1)})}{k_{sj}} = \min_i \frac{(c_{ij} - \lambda_i^{(1)})}{k_{ij}}.$$

Si le minimum est atteint avec plusieurs indices, on opte pour n'importe lequel d'entre eux. Ainsi

$$x_{ij}^{(\nu*)} = \begin{cases} \dfrac{b_j}{k_{ij}} & \text{avec} \quad i = s(j), \\ 0 & \text{avec} \quad i \neq s(j). \end{cases}$$

Le programme de base $X_{\nu*}$ détermine les composantes du vecteur

$$\overline{P}_k = \overline{P}_{\nu*} = \begin{pmatrix} P_\nu^* \\ 1 \end{pmatrix}$$

à ajouter à la base du Z-problème. Le critère du vecteur $\overline{P}_{\nu*}$ par rapport à la base courante s'écrit

$$\Delta_{\nu*} = \lambda^{(2)} - L_\Lambda (X_{\nu*}).$$

Les conditions $\Delta_{\nu*} = 0$ et $\Lambda^{(1)} \leqslant 0$ constituent le critère d'optimalité du programme de base étudié du Z-problème.

Conformément aux remarques du paragraphe 2.1 [cf. la relation (2.32)], la nécessité de construire et de résoudre le X_Λ-problème disparaît aux itérations de la résolution du Z-problème où parmi les composantes $\lambda_i^{(1)}$ du vecteur $\Lambda^{(1)}$ apparaissent des grandeurs positives. Dans ce cas, on ajoute à la base successive du Z-problème le vecteur supplémentaire correspondant.

La résolution du Z-problème permet de calculer le programme optimal du problème de distribution

$$x_{ij}^* = \sum_{r=1}^{m+1} x_{ij}^{(\nu_r)} z_{\nu_r}^*, \qquad i = 1, 2, \ldots, m; \qquad j = 1, 2, \ldots, n,$$

où $\nu_1, \nu_2, \ldots, \nu_{m+1}$ sont les numéros des vecteurs de la base optimale du Z-problème.

La base initiale du Z-problème est déterminée conformément aux observations du paragraphe 1.4.

Supprimons maintenant la contrainte associée à la restriction $k_{ij} > 0$ pour tout i et j. Les coefficients k_{ij} du problème de distribution, de par leur sens physique, peuvent pour certains couples (i, j) devenir des zéros. Dans ce cas, le polyèdre convexe déterminé par les conditions (9.25)-(9.26) devient non borné. C'est pourquoi la condition (9.29) du Z-problème doit être récrite sous la forme

$$\sum_{\nu=1}^{N} \varepsilon_\nu z_\nu = 1,$$

où ε_ν est égal à l'unité si X_ν correspond à un sommet de l'ensemble convexe, et à zéro si X_ν est le vecteur directeur d'une arête non bornée.

Toute collection de couples d'indices $(s(j), j)$, $j = 1, 2, \ldots, n$ tels que $k_{sj} > 0$ correspond à un sommet X_ν de l'ensemble convexe (9.25)-(9.26). Les coordonnées du sommet X_ν sont

$$x_{ij}^{(\nu)} = \begin{cases} \dfrac{b_j}{k_{ij}} & \text{avec} \quad i = s(j), \\ 0 & \text{avec} \quad i \neq s(j). \end{cases}$$

Chaque couple fixe d'indices (s, r) tel que $k_{sr} = 0$ correspond au vecteur directeur d'une arête non bornée de l'ensemble convexe. Les composantes $x_{ij}^{(\nu)}$ du vecteur directeur X_ν s'écrivent

$$x_{ij}^{\nu} = \begin{cases} 1 \text{ avec } i = s, \ j = r, \\ 0 \text{ dans les autres cas.} \end{cases}$$

Le vecteur des conditions $\overline{P}_\nu$ du Z-problème correspondant au sommet X_ν est

$$\overline{P}_\nu = \begin{pmatrix} P_\nu \\ 1 \end{pmatrix},$$

où les composantes de P_ν sont

$$p_\nu^{(i)} = \sum_{j=1}^{n} x_{ij}^{(\nu)}.$$

Le vecteur des conditions $\overline{P}_\nu$ qui correspond à l'arête non bornée associée au couple d'indices (s, r) est égal à

$$\overline{P}_\nu = \begin{pmatrix} P_\nu \\ 0 \end{pmatrix} = \begin{pmatrix} e_s \\ 0 \end{pmatrix},$$

où e_s est le vecteur unité à unité à la s-ième place.

A chaque itération, on calcule le vecteur des conditions du Z-problème ayant le critère maximal.

Parmi les vecteurs $\overline{P}_\nu$ correspondant à des sommets, le plus grand critère $\Delta'_{\nu*}$ revient au vecteur $\overline{P}'_{\nu*}$ tel que

$$x_{ij}^{(\nu*)} = \begin{cases} \dfrac{b_j}{k_{ij}} & \text{avec} \quad i = s(j), \\ 0 & \text{avec} \quad i \neq s(j). \end{cases}$$

Pour tout j, l'indice $s(j)$ se calcule d'après la relation

$$\frac{c_{sj} - \lambda_s^{(1)}}{k_{sj}} = \min_{k_{ij} > 0} \frac{c_{ij} - \lambda_i^{(1)}}{k_{ij}}.$$

La valeur du critère $\Delta'_{\nu*}$ est donnée par la formule

$$\Delta'_{\nu*} = \lambda^{(2)} - \sum_{j=1}^{n} \frac{c_{s(j),\, j} - \lambda_s^{(1)}(j)}{k_{s(j),\, j}} b_j.$$

Parmi les vecteurs correspondant à des arêtes non bornées, le vecteur $\overline{P}''_{\nu*}$ au critère maximal $\Delta''_{\nu*}$ est tel que

$$x_{ij}^{(\nu*)} = \begin{cases} 1, & (i, j) = (s, r), \\ 0, & (i, j) \neq (s, r). \end{cases}$$

Le couple d'indices (s, r) est défini par la relation

$$c_{sr} - \lambda_s^{(1)} = \min_{k_{ij}=0} (c_{ij} - \lambda_i^{(1)}).$$

La valeur du critère $\Delta''_{\nu*}$ est égale à

$$\Delta''_{\nu*} = -L\,(X''_{\nu*}) = -c_{sr} + \lambda_s^{(1)}.$$

Le vecteur à inclure dans la base du Z-problème successif est celui qui a le plus grand critère positif

$$\Delta_{\nu*} = \max\,(\Delta'_{\nu*},\ \Delta''_{\nu*}).$$

Si $\Delta_{\nu*} = \Delta'_{\nu*} > 0$, on adjoint à la base le vecteur $P'_{\nu*}$ correspondant à un sommet; si par contre $\Delta_{\nu*} = \Delta''_{\nu*} > 0$, on inclut dans la base le vecteur $P''_{\nu*}$ associé à une arête non bornée.

9.6. On a déjà dit que la méthode de décomposition et les autres méthodes de programmation par blocs étaient particulièrement efficaces pour la résolution de problèmes de programmation linéaire dans lesquels, en plus des conditions du type transport ou distribution [conditions du type (9.2), (9.3) ou (9.25)], on trouvait une ou plusieurs contraintes de type général. Dans ce cas, les méthodes usuelles de résolution des problèmes de transport ou de distribution ne peuvent pas être employées directement. Les complications du schéma de calcul, suscitées par les conditions supplémentaires de type général, croissent très vite avec l'accroissement du nombre de ces conditions. La méthode de décomposition permet d'éviter ces difficultés. Les contraintes supplémentaires sont incluses dans le bloc à matrice des conditions $A^{(0)}$. Lorsqu'on utilise la méthode de décomposition, les conditions supplémentaires ne compliquent que légèrement la formation des vecteurs des conditions et des coefficients de la forme linéaire du Z-problème, ainsi que le calcul de son programme de base initial.

Le programme optimal du X_Λ-problème, utilisé pour construire le critère d'optimalité du Z-problème, se calcule directement, de même que pour le problème de transport ou de distribution, sans appliquer les méthodes standard de programmation linéaire.

En utilisant les principes de construction d'autres méthodes de programmation linéaire exposés aux §§ 4, 5, 8, on peut construire tout un groupe de schémas de calcul pour l'exploration par blocs des problèmes de distribution et des problèmes similaires (cf. exercice 23).

Voici encore un problème du type transport pour lequel les méthodes de décomposition en blocs peuvent s'avérer efficaces.

Souvent, dans les problèmes de répartition de la production et dans d'autres problèmes économiques et techniques pouvant être interprétés en termes de problème de transport, les volumes de production aux points de production ne sont pas fixés d'avance, mais doivent être déterminés. Ceci étant, ils sont généralement liés par des égalités ou des inégalités linéaires qui constituent chacune telle ou telle contrainte de choix des volumes de production. On suppose en outre connus les coûts liés aux différents volumes de production.

Ce problème se réduit au calcul des variables x_{ij} et x_i qui minimisent la forme linéaire

$$\sum_{j=1}^{n} \sum_{i=1}^{m} c_{ij}x_{ij} + \sum_{i=1}^{m} \sigma_i x_i \tag{9.31}$$

sous les conditions

$$\sum_{i=1}^{m} a_{is}x_i \leqslant a_s, \quad s=1, 2, \ldots, l, \tag{9.32}$$

$$\sum_{j=1}^{n} x_{ij} - x_i = 0, \quad i=1, 2, \ldots, m, \tag{9.33}$$

$$\sum_{i=1}^{m} x_{ij} = b_j, \quad j=1, 2, \ldots, n, \tag{9.34}$$

$$x_{ij} \geqslant 0, \quad i = 1, 2, \ldots, m; \quad j = 1, 2, \ldots, n. \tag{9.35}$$

Désignons comme auparavant par $X_\nu = \{x_{ij}^{(\nu)}\}$ ($\nu = 1, 2, \ldots, N$) les programmes de base du système de conditions (9.34)-(9.35). Comme nous l'avons vu, le calcul des X_ν ne présente aucune difficulté.

Le Z-problème associé au problème (9.31)-(9.35) consiste à calculer les valeurs des variables z_ν et x_i qui minimisent la forme linéaire

$$\sum_{\nu=1}^{N} c_\nu z_\nu + \sum_{i=1}^{m} \sigma_i x_i \tag{9.36}$$

sous les conditions

$$\sum_{i=1}^{m} a_{is}x_i \leqslant a_s, \quad s=1, 2, \ldots, l, \tag{9.37}$$

$$\sum_{\nu=1}^{N} \sum_{j=1}^{n} x_{ij}^{(\nu)} z_\nu - x_i = 0, \quad i=1, 2, \ldots, m, \tag{9.38}$$

$$\sum_{\nu=1}^{N} z_\nu = 1, \tag{9.39}$$

$$z_\nu \geqslant 0, \quad \nu = 1, 2, \ldots, N. \tag{9.40}$$

En éliminant x_i des conditions (9.37), (9.38) et de la forme linéaire (9.36), on ramène le Z-problème à la forme

$$\sum_{\nu=1}^{N} \tilde{c}_\nu z_\nu \to \min$$

sous les conditions

$$\sum_{\nu=1}^{N} \tilde{a}_s^{(\nu)} z_\nu \leqslant a_s, \quad s=1, 2, \ldots, l,$$

$$\sum_{\nu=1}^{N} z_\nu = 1,$$

$$z_\nu \geqslant 0, \quad \nu = 1, 2, \ldots, N,$$

où

$$\tilde{c}_\nu = \sum_{i=1}^{m} \sum_{j=1}^{n} (c_{ij} + \sigma_i) x_{ij}^{(\nu)},$$

$$\tilde{a}_s^{(\nu)} = \sum_{i=1}^{m} \sum_{j=1}^{n} a_{is} x_{ij}^{(\nu)}.$$

Les recommandations pratiques pour l'application de la méthode de décomposition au problème (9.35)-(9.36) se déduisent des remarques faites aux paragraphes précédents dans la description des principes d'utilisation de cette méthode pour la résolution des problèmes de transport et de distribution.

§ 10. Problèmes de transport à indices multiples

Les méthodes de programmation par blocs simplifient dans de nombreux cas la résolution des problèmes de transport à indices multiples.

Considérons l'application de la méthode de décomposition à la résolution des problèmes de transport à trois indices du type suivant [13].

Soit la forme linéaire à minimiser

$$L(X) = \sum_{i_3=1}^{n_3} \sum_{i_2=1}^{n_2} \sum_{i_1=1}^{n_1} c_{i_1 i_2 i_3} x_{i_1 i_2 i_3} \tag{10.1}$$

sous les conditions

$$\sum_{i_2=1}^{n_2} \sum_{i_1=1}^{n_1} x_{i_1 i_2 i_3} = a_{i_3}^{(3)}, \quad i_3 = 1, 2, \ldots, n_3, \tag{10.2}$$

$$\sum_{i_3=1}^{n_3} \sum_{i_1=1}^{n_1} x_{i_1 i_2 i_3} = a_{i_2}^{(2)}, \quad i_2 = 1, 2, \ldots, n_2, \tag{10.3}$$

$$\sum_{i_3=1}^{n_3} \sum_{i_2=1}^{n_2} x_{i_1 i_2 i_3} = a_{i_1}^{(1)}, \quad i_1 = 1, 2, \ldots, n_1, \tag{10.4}$$

$$x_{i_1 i_2 i_3} \geqslant 0 \quad \text{pour tous les } i_1, i_2, i_3. \tag{10.5}$$

Le problème (10.1)-(10.5) s'appelle *problème à trois indices à sommes axiales.*

La condition de résolubilité de ce problème est

$$\sum_{i_1=1}^{n_1} a_{i_1}^{(1)} = \sum_{i_2=1}^{n_2} a_{i_2}^{(2)} = \sum_{i_3=1}^{n_3} a_{i_3}^{(3)} = T. \tag{10.6}$$

La collection des variables

$$x_{i_1 i_2 i_3} = \frac{a_{i_1}^{(1)} a_{i_2}^{(2)} a_{i_3}^{(3)}}{T^2} \tag{10.7}$$

constitue le programme du problème.

Soient $n_3 \leqslant n_2$ et $n_3 \leqslant n_1$.

Considérons l'ensemble convexe M défini par les conditions (10.3)-(10.5). On voit aisément qu'il est borné. Désignons par X_ν ($\nu = 1, 2, \ldots, N$) les sommets du polyèdre M. Tout point du polyèdre M peut s'écrire

$$X = \sum_{\nu=1}^{N} X_\nu z_\nu \tag{10.8}$$

ou être mis en coordonnées sous la forme

$$x_{i_1 i_2 i_3} = \sum_{\nu=1}^{N} x_{i_1 i_2 i_3}^{(\nu)} z_\nu, \tag{10.9}$$

où

$$\sum_{\nu=1}^{N} z_\nu = 1, \tag{10.10}$$

$$z_\nu \geqslant 0, \quad \nu = 1, 2, \ldots, N. \tag{10.11}$$

La forme linéaire (10.1) et les contraintes (10.3)-(10.5) définissent le X'-problème. La collection des vecteurs X_ν ($\nu = 1, 2, \ldots, N$) est l'ensemble des programmes de base du X'-problème.

Le Z-problème associé au X-problème à trois indices initial consiste à minimiser la forme linéaire

$$\sum_{\nu=1}^{N} c_\nu z_\nu \tag{10.12}$$

sous les conditions

$$\sum_{\nu=1}^{N} p_\nu^{(i_3)} z_\nu = a_{i_3}^{(3)}, \quad i_3 = 1, 2, \ldots, n_3, \tag{10.13}$$

$$z_\nu \geqslant 0, \quad \nu = 1, 2, \ldots, N, \tag{10.14}$$

où

$$c_\nu = \sum_{i_3=1}^{n_3} \sum_{i_2=1}^{n_2} \sum_{i_1=1}^{n_1} = c_{i_1 i_2 i_3} x_{i_1 i_2 i_3}^{(\nu)}, \tag{10.15}$$

$$p_\nu^{(i_3)} = \sum_{i_2=1}^{n_2} \sum_{i_1=1}^{n_1} x_{i_1 i_2 i_3}^{(\nu)}. \tag{10.16}$$

L'équivalence du Z-problème et du X-problème est facile à vérifier de la même façon que pour le problème de transport classique. L'égalité (10.10) se déduit des équations (10.13) et des conditions (10.6) de résolubilité du problème initial.

Soient $\{x_{i_1 i_2}^{(t)}\}$, $t = 1, 2, \ldots, n_3$, des programmes de base arbitraires du problème de transport aux volumes de production et de consommation $a_{i_1}^{(1)}$ $(i_1 = 1, 2, \ldots, n_1)$, $a_{i_2}^{(2)}$ $(i_2 = 1, 2, \ldots, n_2)$.

Choisissons n_3 programmes de base du X'-problème aux composantes

$$x_{i_1 i_2 i_3}^{(\nu_t)} = \begin{cases} x_{i_1 i_2}^{(t)} & \text{avec} \quad i_3 = t, \\ 0 & \text{avec} \quad i_3 \neq t, \end{cases} \quad t = 1, 2, \ldots, n_3. \tag{10.17}$$

A chacun de ces programmes correspond un vecteur des conditions du Z-problème aux composantes [cf. (10.16)]

$$p_{\nu_t}^{(i_3)} = \begin{cases} T & \text{avec} \quad i_3 = t, \\ 0 & \text{avec} \quad i_3 \neq t, \end{cases} \quad t = 1, 2, \ldots, n_3. \tag{10.18}$$

Les vecteurs P_{ν_t} peuvent être pris comme base initiale du Z-problème. A cette base correspond le programme initial de base du Z-problème aux composantes

$$z_{\nu_t} = \frac{a_t^{(3)}}{T}, \quad t = 1, 2, \ldots, n_3, \tag{10.19}$$

Les coefficients respectifs de la forme linéaire du Z-problème sont [cf. (10.15)]

$$c_{\nu_t} = \sum_{i_2=1}^{n_2} \sum_{i_1=1}^{n_1} c_{i_1 i_2 t} x_{i_1 i_2}^{(t)}, \quad t = 1, 2, \ldots, n_3. \tag{10.20}$$

On voit que pour des t différents, il est rationnel de prendre des programmes $x_{i_1 i_2}^{(t)}$ différents. Le choix des $x_{i_1 i_2}^{(t)}$ peut s'effectuer en minimisant $\sum\sum c_{i_1 i_2 t} x_{i_1 i_2}$ par une méthode exacte ou approchée.

La matrice de la base est une matrice unité multipliée par le scalaire T. Sa matrice inverse est une matrice unité divisée par T. C'est pourquoi les composantes λ_t du vecteur Λ des critères des conditions du Z-problème par rapport à sa base initiale se calculent d'après la formule

$$\lambda_t = \frac{c_{\nu_t}}{T}, \quad t = 1, 2, \ldots, n_3. \tag{10.21}$$

Le X_Λ-problème dont la résolution sert à tester si le programme de base du Z-problème est optimal prend la forme suivante.

Calculer les variables $x_{i_1 i_2 i_3}$ qui minimisent la forme linéaire

$$L_\Lambda(X) = \sum_{i_3=1}^{n_3} \sum_{i_2=1}^{n_2} \sum_{i_1=1}^{n_1} (c_{i_1 i_2 i_3} - \lambda_{i_3}) x_{i_1 i_2 i_3} \tag{10.22}$$

sous les conditions

$$\sum_{i_3=1}^{n_3} \sum_{i_1=1}^{n_1} x_{i_1 i_2 i_3} = a_{i_2}^{(2)}, \quad i_2 = 1, 2, \ldots, n_2, \tag{10.23}$$

$$\sum_{i_3=1}^{n_3} \sum_{i_2=1}^{n_2} x_{i_1 i_2 i_3} = a_{i_1}^{(1)}, \quad i_1 = 1, 2, \ldots, n_1, \tag{10.24}$$

$$x_{i_1 i_2 i_3} \geqslant 0, \quad i_1 = 1, 2, \ldots, n_1; \quad i_2 = 1, 2, \ldots, n_2; \quad i_3 = 1, 2, \ldots, n_3. \tag{10.25}$$

Montrons que la résolution du X_Λ-problème (10.22)-(10.25) se ramène au calcul du programme optimal d'un problème de transport classique de dimensions $n_1 \times n_2$.

Fixons l'indice i_3, pour les valeurs données des indices i_1 et i_2, à l'aide de la condition

$$c_{i_1 i_2} = c_{i_1 i_2 s(i_1, i_2)} - \lambda_{s(i_1, i_2)} = \min_{i_3} (c_{i_1 i_2 i_3} - \lambda_{i_3}). \tag{10.26}$$

Si le minimum est atteint pour plusieurs valeurs de l'indice i_3, on choisit l'une quelconque d'entre elles.

Considérons le problème de transport classique qui consiste à calculer les variables $y_{i_1 i_2}$ ($i_1 = 1, 2, \ldots, n_1$; $i_2 = 1, 2, \ldots, n_2$) minimisant la forme linéaire

$$\sum_{i_2=1}^{n_2} \sum_{i_1=1}^{n_1} c_{i_1 i_2} y_{i_1 i_2} \tag{10.27}$$

sous les conditions

$$\sum_{i_1=1}^{n_1} y_{i_1 i_2} = a_{i_2}^{(2)}, \quad i_2 = 1, 2, \ldots, n_2, \tag{10.28}$$

$$\sum_{i_2=1}^{n_2} y_{i_1 i_2} = a_{i_1}^{(1)}, \quad i_1 = 1, 2, \ldots, n_1, \tag{10.29}$$

$$y_{i_1 i_2} \geqslant 0, \quad i_1 = 1, 2, \ldots, n_1; \quad i_2 = 1, 2, \ldots, n_2. \tag{10.30}$$

Supposons que la collection de nombres $y^*_{i_1 i_2}$ ($i_1 = 1, 2, \ldots, n_1$; $i_2 = 1, 2, \ldots, n_2$) définisse la solution du problème de transport (10.27)-(10.30). Alors, les quantités

$$x^*_{i_1 i_2 i_3} = \begin{cases} y^*_{i_1 i_2} & \text{avec} \quad i_3 = s(i_1, i_2), \\ 0 & \text{avec} \quad i_3 \neq s, \end{cases} \tag{10.31}$$

$$i_1 = 1, 2, \ldots, n_1; \quad i_2 = 1, 2, \ldots, n_2; \quad i_3 = 1, 2, \ldots, n_3,$$

sont les valeurs des composantes du programme de base optimal du X_Λ-problème.

En effet, par substitution directe de (10.31) dans les conditions (10.23)-(10.25), on constate que la collection de nombres $x^*_{i_1i_2i_3}$ est un programme du X_Λ-problème (un programme de base, comme on l'établit aisément). De plus, il est évident que

$$L_\Lambda(X) = \sum_{i_3=1}^{n_3} \sum_{i_2=1}^{n_2} \sum_{i_1=1}^{n_1} (c_{i_1i_2i_3} - \lambda_{i_3})\, x_{i_1i_2i_3} \geqslant \sum_{i_2=1}^{n_2} \sum_{i_1=1}^{n_1} c_{i_1i_2} y_{i_1i_2},$$

où $y_{i_1i_2} = \sum_{i_3=1}^{n_3} x_{i_1i_2i_3}$ est un programme du problème (10.27)-(10.30) et $c_{i_1i_2}$ est défini par la condition (10.26).

Ensuite,

$$\sum_{i_2=1}^{n_2} \sum_{i_1=1}^{n_1} c_{i_1i_2} y_{i_1i_2} \geqslant \sum_{i_2=1}^{n_2} \sum_{i_1=1}^{n_1} c_{i_1i_2} y^*_{i_1i_2} =$$

$$= \sum_{i_3=1}^{n_3} \sum_{i_2=1}^{n_2} \sum_{i_1=1}^{n_1} (c_{i_1i_2i_3} - \lambda_{i_3})\, x^*_{i_1i_2i_3} \text{ [cf. (10.31)]}.$$

Donc,

$$L_\Lambda(X) \geqslant L_\Lambda(X^*) = L^*_\Lambda$$

pour tout programme X du problème (10.22)-(10.25), ce qui signifie que X^* est solution de ce problème.

La valeur L^*_Λ de la forme linéaire du X_Λ-problème sur son programme optimal $X_{\nu*}$ est de module égal et de signe opposé au critère du vecteur des conditions $P_{\nu*}$ par rapport à la base courante du Z-problème :

$$\Delta_{\nu*} = -L^*_\Lambda = -\sum_{i_2=1}^{n_2} \sum_{i_1=1}^{n_1} c_{i_1i_2} y^*_{i_1i_2}.$$

L'égalité $\Delta_{\nu*} = 0$ témoigne de ce que la base du Z-problème étudiée correspond à son programme optimal. La résolution du problème à trois indices initial se calcule dans ce cas selon la formule (10.8), où z_ν sont les composantes du programme optimal du Z-problème.

L'inégalité $\Delta_{\nu*} > 0$ signifie qu'il faut poursuivre la procédure de résolution conformément aux règles générales de la méthode primale du simplexe. Le Z-problème et avec lui le problème de transport à trois indices initial sont résolus en un nombre fini d'itérations.

De cette façon, la résolution du problème de transport à trois indices est ramenée à la résolution d'un problème de programmation linéaire de type général à n_3 contraintes et d'une suite de problèmes de transport classiques de dimensions $n_1 \times n_2$. De plus, il est avantageux de prendre comme programme initial de chacune des itérations successives du problème de transport la solution du X_Λ-problème précédent.

Exactement de la même manière, il est possible de ramener le problème de transport à quatre indices à sommes axiales (qui impose la sommation suivant trois indices) à un problème de programmation linéaire de type général et à une suite de problèmes de transport à trois indices. De façon analogue, le problème de transport à t indices à sommes axiales se ramène à un problème de programmation linéaire de type général et à une suite de problèmes de transport à $(t-1)$ indices à sommes axiales.

Dans les problèmes de transport pratiques à indices multiples, il arrive souvent que la marge de variation d'un ou de plusieurs indices soit très petite par rapport à la marge de variation des autres indices. Dans de tels cas, l'application de la programmation par blocs s'avère très efficace.

Les méthodes de programmation par blocs exposées aux §§ 4, 5, 8 peuvent servir de base pour construire d'autres schémas de calcul des problèmes de transport à indices multiples. Une appréciation comparative des algorithmes de résolution du problème de transport à indices multiples, fondés sur les différentes méthodes de programmation par blocs, ne peut être effectuée qu'après avoir accumulé une certaine expérience des calculs.

§ 11. Une application de la méthode de décomposition

Il existe plusieurs problèmes pratiques où tous ou une partie des vecteurs des conditions ne sont pas fixés d'avance, mais seulement limités par un système d'égalités ou d'inégalités linéaires. Chacun de ces vecteurs des conditions appartient à un certain ensemble convexe. Le problème consiste ainsi à choisir les paramètres de commande x_j et les vecteurs des conditions $a^{(j)} \in M_j$ succeptibles d'optimiser le critère imposé de la qualité de la résolution. On sous-entend ici par M_j un ensemble convexe, donné par un système d'égalités et d'inégalités linéaires.

En termes formels, le problème se ramène au calcul des variables x_j et des vecteurs $a^{(j)}$ qui maximisent la forme linéaire

$$L = \sum_{j=1}^{n} c_j x_j \tag{11.1}$$

sous les conditions

$$\sum_{j=1}^{n} a^{(j)} x_j = b^{(0)}, \tag{11.2}$$

$$A^{(j)} a^{(j)} \leqslant b^{(j)}, \quad j = 1, 2, \ldots, n, \tag{11.3}$$

$$x_j \geqslant 0, \quad j = 1, 2, \ldots, n. \tag{11.4}$$

$A^{(j)}$ est ici la matrice de dimension $m_j \times m$; $b^{(0)}$, le vecteur de dimension m; $b^{(j)}$, le vecteur de dimension m_j; $a^{(j)}$, le vecteur de dimension m.

La méthode de résolution du problème (11.2)-(11.4) décrite ci-dessous est exposée dans [35].

Considérons le cas où les domaines de définition des conditions (11.3) sont bornés et où, par conséquent, les ensembles convexes M_j sont des polyèdres convexes. Le rejet de cette hypothèse entraîne certaines modifications négligeables de la méthode de résolution du problème. En tenant compte des remarques du paragraphe 1.3, le lecteur pourra les réaliser lui-même.

Désignons les sommets du polyèdre M_j par $a_\nu^{(j)}$ ($\nu = 1, 2, \ldots, s_j$). Chaque point $a^{(j)} \in M_j$ peut être représenté comme une combinaison convexe des sommets du polyèdre M_j:

$$a^{(j)} = \sum_{\nu=1}^{s_j} a_\nu^{(j)} z_\nu^{(j)}, \quad j = 1, 2, \ldots, n_1, \tag{11.5}$$

où

$$\sum_{\nu=1}^{s_j} z_\nu^{(j)} = 1, \quad j = 1, 2, \ldots, n, \tag{11.6}$$

$$z_\nu^{(j)} \geqslant 0, \quad \nu = 1, 2, \ldots, s_j; \quad j = 1, 2, \ldots, n. \tag{11.7}$$

Les conditions (11.6) permettent de mettre (11.1) sous la forme

$$\widetilde{L} = \sum_{j=1}^{n} \sum_{\nu=1}^{s_j} c_j x_j z_\nu^{(j)}. \tag{11.8}$$

En tenant compte de l'expression (11.5) pour $a^{(j)}$ le problème étudié peut être reformulé de la façon suivante.

Calculer les variables x_j et $z_\nu^{(j)}$ ($\nu = 1, 2, \ldots, s_j$; $j = 1, 2, \ldots, n$) qui maximisent la fonction (11.8) sous les conditions

$$\sum_{j=1}^{n} \sum_{\nu=1}^{s_j} a_\nu^{(j)} x_j z_\nu^{(j)} = b^{(0)}, \tag{11.9}$$

$$\sum_{\nu=1}^{s_j} z_\nu^{(j)} = 1, \quad j = 1, 2, \ldots, n, \tag{11.10}$$

$$x_j \geqslant 0; \quad z_\nu^{(j)} \geqslant 0; \quad \nu = 1, 2, \ldots, s_j; \quad j = 1, 2, \ldots, n. \tag{11.11}$$

Introduisons les nouvelles variables

$$y_\nu^{(j)} = x_j z_\nu^{(j)}, \quad \nu = 1, 2, \ldots, s_j; \quad j = 1, 2, \ldots, n. \tag{11.12}$$

Les variables $y_\nu^{(j)}$ astreignent le problème à maximiser la forme linéaire

$$\widetilde{L} = \sum_{j=1}^{n} \sum_{\nu=1}^{s_j} c_j y_\nu^{(j)} \tag{11.13}$$

sous les conditions

$$\sum_{j=1}^{n} \sum_{\nu=1}^{s_j} a_\nu^{(j)} y_\nu^{(j)} = b^{(0)}, \tag{11.14}$$

$$y_\nu^{(j)} \geqslant 0, \quad \nu = 1, 2, \ldots, s_j; \quad j = 1, 2, \ldots, n. \tag{11.15}$$

Pour rendre la résolution de ce problème réalisable il suffit simplement de connaître les coordonnées de m sommets des polyèdres M_j, qui correspondent à la base initiale du problème (11.13)-(11.15). Ces sommets peuvent appartenir à l'un ou aux différents polyèdres M_j.

Le critère d'optimalité du programme de base du problème (11.13)-(11.15) est

$$\Delta_\nu^{(j)} = (\Lambda, a_\nu^{(j)}) - c_j \geqslant 0 \quad \text{pour tout } j \text{ et } \nu. \tag{11.16}$$

Λ est ici le vecteur des critères des conditions du problème (11.13)-(11.15) par rapport à la base donnée.

Au lieu de tester toutes les conditions (11.16), ce qui est une tâche irréalisable lorsque les s_j sont grands, il suffit d'évaluer les valeurs optimales des formes linéaires de la série suivante de problèmes de programmation linéaire:

$$c_j - (\Lambda, a^{(j)}) \to \max \tag{11.17}$$

$$\text{avec } a^{(j)} \in M_j, \quad j = 1, 2, \ldots, n, \tag{11.18}$$

ou, ce qui revient au même,

$$(\Lambda, a^{(j)}) \to \min$$

$$\text{avec } a^{(j)} \in M_j, \quad j = 1, 2, \ldots, n.$$

La base considérée du problème (11.13)-(11.15) est optimale si les valeurs optimales des fonctions économiques (11.17) sont nulles pour tout j. Si parmi les valeurs optimales des fonctions (11.17) il y a des nombres positifs, la procédure de résolution du problème (11.13)-(11.15) se poursuit, et l'on ajoute à la base le vecteur $a_{\nu*}^{(h)}$ tel que

$$(\Lambda, a_{\nu*}^{(h)}) = \min_j (\Lambda, a_{\nu*}^{(j)}).$$

$a_{\nu*}^{(j)}$ est ici une solution du problème (11.17)-(11.18).

La résolution du problème (11.13)-(11.15) définit aussi bien les vecteurs des conditions rationnels que le programme optimal du problème initial.

Les relations (11.12) et (11.10) donnent

$$x_j^* = \sum_{\nu=1}^{s_j} y_\nu^{*(j)}, \quad j = 1, 2, \ldots, n.$$

Les valeurs optimales des vecteurs des conditions $a^{(j)}$ qui correspondent aux composantes positives de la solution du problème ($x_j^* > 0$) se calculent d'après la formule

$$a^{*(j)} = \sum_{\nu=1}^{s_j} a_\nu^{(j)} z_\nu^{*(j)},$$

où

$$z_\nu^{*(j)} = \frac{y_\nu^{*(j)}}{x_j^*},$$

et où $y_\nu^{*(j)}$ sont les composantes de la solution du problème (11.13)-(11.15).

Les vecteurs des conditions $a^{(j)}$ non inclus dans la base du programme optimal du problème (11.1)-(11.4) ne se calculent naturellement pas. Les valeurs de leurs composantes ne présentent aucun intérêt pratique.

Exercices

1. Composer le schéma bloc de l'algorithme de décomposition:
a) pour le problème de type (1.4)-(1.7);
b) pour le problème de type (2.1)-(2.4);
c) pour le problème de transport et le problème de distribution;
d) pour le problème de transport à indices multiples à sommes axiales.

2. Composer les schémas de calcul pour les modalités de la méthode de décomposition décrites au § 2.

3. Donner la description de l'algorithme de la méthode de décomposition appliqué au problème borné inférieurement et supérieurement (cf. [87], § 4 du chap. 11).

4. Appliquer le principe de décomposition à la résolution du problème de programmation linéaire à matrice des conditions du paragraphe 2.6. Donner la description du schéma de calcul.

5. Donner la description d'un schéma de calcul pour l'analogue dual de la méthode de décomposition (cf. § 4) appliqué au problème à matrice des conditions du paragraphe 2.6.

6. Donner une interprétation économique de la programmation par blocs en termes de problème de la combinaison rationnelle des fonctions des organes locaux et centraux de planification.

7. Montrer qu'avec $X \in G_\Lambda$, l'hyperplan (3.34) est pour l'ensemble $\overline{R}$ un hyperplan d'appui en $\overline{\Lambda}'$.

8. Montrer que pour que $\Lambda' + \varepsilon l \in R$ avec $\Lambda' \in R$ et $\varepsilon > 0$ petit, il faut et il suffit que le deuxième membre de (3.35) soit borné supérieurement.

9. Démontrer que si $(\Lambda, \lambda^{(1)})$ est le vecteur des critères des conditions du Z-problème par rapport à une base donnée $(\overline{P}_{i_1}, \overline{P}_{i_2}, \ldots, \overline{P}_{i_{m+1}})$,

$$\overline{\Lambda} = (\Lambda, \lambda), \quad \text{où} \quad \lambda = \lambda^{(1)} + (B^{(0)}, \Lambda),$$

est alors le point commun aux hyperplans H_{i_s}, $s = 1, 2, \ldots, m+1$.

10. Montrer la validité de la formule (4.8).

11. Démontrer l'équivalence des problèmes $\Gamma(M, \Lambda')$ et (5.6)-(5.9).

12. Justifier la relation (5.42).

13. Pour choisir la direction possible, maximiser la fonction $\varphi_2(X)$ en définissant l'ensemble M par les conditions

$$l_i \geqslant -1, \quad i = 1, 2, \ldots, m,$$

et pour calculer le déplacement dans la direction possible utiliser le mode E_0. Démontrer que la méthode de programmation par blocs ainsi obtenue est associée à la résolution du Z-problème par la méthode primale-duale.

14. Justifier la méthode de résolution d'un problème décomposé en blocs associée à l'introduction de la contrainte supplémentaire (6.1) (cf. le paragraphe 6.1).

15. Démontrer que les conditions du cas b) du paragraphe 6.4 sont équivalentes à la relation (6.10).

16. Montrer que si la relation (7.11) est vérifiée

$$(C, X'_\varepsilon) \geqslant (C, X^*) - \alpha\varepsilon,$$

où X^* est la solution du problème (3.1)-(3.4), le coefficient α ne dépend pas de Λ' et $\varepsilon > 0$.

17. Trouver une expression pour la fonction $f(\Lambda)$ lorsque la matrice $A^{(1)}$ des conditions (3.3) est du type diagonal par blocs.

18. Comparer les volumes de calcul imposés par les différentes méthodes du problème ci-dessus (cf. §§ 1 à 8).

Maximiser la forme linéaire

$$L(X) = 2x_1 - x_2 + 5x_3 + 2x_5 - 3x_6 - x_7 + 4x_8$$

sous les conditions

$$\begin{aligned}
3x_1 + 4x_2 + x_3 + 2x_4 + x_5 + x_6 + 5x_7 + 2x_8 &\leqslant 20,\\
-x_1 + 2x_2 + x_4 + 2x_5 + 5x_6 + 3x_7 - x_8 &\leqslant 15,\\
2x_1 + x_2 + x_3 &\leqslant 8,\\
-3x_1 + 2x_3 &\leqslant 6,\\
3x_4 + 4x_5 &\leqslant 10,\\
-x_4 + x_5 &\leqslant 4,\\
2x_4 + 3x_5 &\leqslant 9,\\
x_6 + 4x_7 + x_8 &\leqslant 12,\\
-3x_6 + x_7 &\leqslant 5,\\
-2x_7 + 4x_8 &\leqslant 7,
\end{aligned}$$

$$x_j \geqslant 0, \quad j = 1, 2, \ldots, 8.$$

19. Résoudre, en utilisant la méthode de décomposition, le problème de transport à matrice des coefficients dont la forme linéaire s'écrit

$$C = \{c_{ij}\} = \left\| \begin{matrix} 4 & 3 & 2 & 6 & 1 & 8 & 3 & 7 & 2 & 4 & 6 & 9 \\ 2 & 7 & 4 & 8 & 3 & 9 & 5 & 5 & 4 & 8 & 1 & 4 \end{matrix} \right\|$$

et les volumes de consommation et de production sont définis par les vecteurs

$$a = (a_1, a_2) = (40; 60),$$

$$b = (b_1, b_2, \ldots, b_{12}) = (7; 5; 12; 3; 10; 5; 9; 1; 15; 8; 11; 14).$$

20. Dresser les schémas de calcul pour la résolution du problème de transport en appliquant les méthodes de programmation par blocs des §§ 4, 5 et 8.

21. Résoudre le problème de l'exercice 19 d'après les schémas de calcul de l'exercice précédent.

22. Trouver à l'aide de la méthode de décomposition le programme optimal du problème de distribution à forme linéaire et aux volumes de production et de consommation du problème de l'exercice 19 et à matrice suivante des coeffi-

cients k_{ij}:

$$K = \{k_{ij}\} = \begin{Vmatrix} 1 & 3 & 1 & 1 & 5 & 1 & 1 & 1 & 2 & 1 & 3 & 2 \\ 4 & 1 & 2 & 1 & 1 & 1 & 2 & 3 & 1 & 2 & 1 & 1 \end{Vmatrix}.$$

23. Construire les schémas de calcul pour résoudre le problème de distribution en partant des méthodes de programmation par blocs des §§ 4, 5, 8.

24. Résoudre le problème de l'exercice 22 suivant les schémas de calcul de l'exercice précédent.

25. Résoudre par la méthode de décomposition le problème de transport à trois indices et à sommes axiales soumis aux conditions suivantes:

$$n_1 = 2; \quad n_2 = 3; \quad n_3 = 15;$$

$$c_{i_1 i_2 i_3}$$

	$i_1 = 1$			$i_1 = 2$		
i_3 \ i_2	1	2	3	1	2	3
1	5	2	7	8	1	6
2	1	6	3	3	5	2
3	4	5	1	4	2	1
4	1	4	8	5	8	4
5	8	3	6	1	2	7
6	5	1	9	3	5	6
7	4	7	3	8	1	3
8	3	5	4	4	6	9
9	8	1	2	8	7	3
10	1	4	7	1	4	1
11	9	6	1	9	2	5
12	5	2	3	2	8	2
13	8	4	6	4	3	6
14	1	9	5	7	4	5
15	3	5	2	1	6	7

$$a_1^{(1)} = 40; \quad a_2^{(1)} = 60;$$

$$a_1^{(2)} = 35; \quad a_2^{(2)} = 25; \quad a_3^{(2)} = 40;$$

$$a_{i_3}^{(3)} = (15;\ 5;\ 4;\ 7;\ 12;\ 8;\ 3;\ 6;\ 4;\ 3;\ 5;\ 4;\ 10;\ 6;\ 8).$$

26. Construire l'algorithme de maximisation de la forme linéaire

$$\sum_{j=1}^{n} c_j x_j$$

sous les conditions

$$\sum_{j=1}^{n} a^{(j)} x_j \leqslant b^{(0)}, \quad x_j \geqslant 0, \qquad j = 1, 2, \ldots, n,$$

si, en ce qui concerne les vecteurs des conditions $a^{(j)}$, on sait seulement que la projection de n'importe lequel d'entre eux sur un axe de coordonnées arbitraire est comprise entre 0 et 1.

Chapitre 5

PROGRAMMATION LINÉAIRE EN NOMBRES ENTIERS

Les problèmes de programmation linéaire rendus plus complexes par une contrainte qui astreint toutes les variables ou certaines d'entre elles d'être entières (problèmes de programmation linéaire en nombres entiers) présentent un grand intérêt pratique. D'une part, les conditions d'intégralité sont caractéristiques de nombreux modèles, applicables aux domaines les plus divers. D'autre part, c'est en termes de programmation en nombres entiers que se formule une série de problèmes extrémaux difficiles dont les méthodes d'analyse sont encore insuffisamment étudiées.

Dans le § 1 nous avons recensé quelques-unes des applications. Les énoncés qui y sont donnés suffisent amplement pour faire sentir les énormes possibilités qui s'ouvrent à mesure que se développent et se perfectionnent les méthodes de résolution des problèmes en nombres entiers.

Dans certains cas, la discussion d'un modèle en nombres entiers n'exige pas de méthodes spéciales du fait que chaque solution du problème linéaire correspondant se soumet automatiquement aux conditions d'intégralité. Il en est ainsi, notamment, pour les problèmes dont les ensembles convexes des conditions n'ont que des sommets en nombres entiers.

Le § 2 est consacré à l'exposé de plusieurs conditions qui permettent de juger de la présence de la propriété mentionnée dans les ensembles convexes donnés à l'aide de systèmes de contraintes linéaires.

Les deux paragraphes suivants donnent la description détaillée et la justification d'une des méthodes de programmation en nombres entiers, due à R. Gomory. Le § 3 examine le cas où toutes les variables du problème sont astreintes à être entières; le § 4 est consacré aux programmes partiellement en nombres entiers.

Le dernier paragraphe expose un autre algorithme de Gomory et une méthode de résolution des problèmes en nombres entiers fondée sur les principes de la programmation dynamique.

A noter particulièrement le numéro 5.3, qui établit la divergence de la procédure de calcul qui a à sa base une règle de formation des contraintes additionnelles proposée par Dantzig.

§ 1. Problèmes de programmation linéaire en nombres entiers

1.1. Il existe de nombreuses formulations de problèmes extrémaux importants et complexes présentant un grand intérêt et appliqués aux domaines les plus divers qui se ramènent à l'analyse de problèmes de programmation linéaire dont toutes ou une partie des variables sont astreintes à être entières.

Appelons problème de programmation linéaire (totalement) *en nombres entiers* (P.L.T.) le problème qui consiste à maximiser (ou à minimiser) une forme linéaire arbitraire

$$(C, X) = \sum_{j=1}^{n} c_j x_j$$

sous des conditions linéaires quelconques du type égalités et inégalités, et soumis à la contrainte additionnelle

$$x_j \text{ entier pour } j = 1, 2, \ldots, n. \tag{1.1}$$

Si les conditions d'intégralité (1.1) ne portent que sur une partie des variables x_j, les modèles correspondants s'appellent problèmes de programmation linéaire *partiellement en nombres entiers* et les méthodes de leur analyse programmation linéaire *partiellement en nombres entiers* (P.L.P.). Pour plus de concision, nous omettons par la suite le terme « linéaire » et parlons de problèmes de programmation en nombres entiers ou partiellement en nombres entiers. Ceci ne peut prêter à confusion, dans la mesure où notre champ d'étude se borne à la classe des problèmes linéaires. L'intérêt qui se manifeste actuellement pour la programmation en nombres entiers s'explique par l'abondance des problèmes pratiques qui cadrent avec le schéma des modèles en nombres entiers. Il faut souligner de plus que les termes de programmation en nombres entiers permettent d'établir la formulation de nombreux problèmes complexes dont les méthodes de résolution ne sont pas encore mises au point.

Le présent paragraphe traite de plusieurs applications caractéristiques de la programmation en nombres entiers, qui illustrent l'importance et la valeur pratique de cette branche d'avenir de programmation mathématique. Les problèmes à considérer appartiennent à trois classes. Ce sont des problèmes qui donnent la description mathématique de phénomènes portant sur des objets indivisibles (usines, wagons, avions, etc.); les problèmes de type com-

binatoire et les problèmes non linéaires qui dépassent le cadre de la programmation convexe.

1.2. Dans certains cas, le sens physique des paramètres de commande (variables du problème) leur impose de ne prendre que des valeurs entières. Plusieurs problèmes de ce type sont donnés dans les chapitres 1-3 et 13 de [87].

Le § 2 du premier chapitre de [87] cite un problème lié au choix d'un système d'armement. On y a pris comme paramètres de commande (variables) les nombres inconnus de collections complexes d'armement de divers types. Il ne pouvait naturellement être question que de collections entières. C'est pourquoi le problème du choix optimal du système d'armement contient dans sa formulation exacte des contraintes du type (1.1).

Le problème de la coupe optimale des matériaux énoncé dans [87] (§ 13 du chapitre 2) est également, en toute rigueur, un problème en nombres entiers, puisque ses paramètres de commande (variables) se mesurent en unités entières (feuilles entières de contre-plaqué, de fer blanc, de verre, etc.). On trouve quelques problèmes pratiques de programmation en nombres entiers au § 2 du chapitre 3 de [87], consacré au problème de distribution. Par exemple, la répartition optimale des avions entre les lignes aériennes se ramène à un problème de distribution dont la signification physique des paramètres de commande (variables) est le nombre d'avions d'un type donné affectés à telle ou telle ligne aérienne. Nous avons donc encore affaire à un problème de programmation en nombres entiers.

Il existe des problèmes dont les paramètres de commande ont un sens logique naturel. Le plus souvent, les paramètres logiques peuvent prendre deux valeurs seulement : 0 ou 1. La première valeur du paramètre répond au cas où telle ou telle proposition n'a pas lieu, la seconde indique que cette affirmation est vraie.

Plusieurs problèmes pratiques, tels que la répartition de l'équipement dans un avion ou une fusée, l'emballage compact de matériel divers, etc., se rangent dans le schéma suivant.

Soient n objets, dont chacun est caractérisé, en fonction de l'application considérée, par un certain coefficient d'utilité. Dans les divers problèmes concrets, ce coefficient peut caractériser différentes propriétés des objets. Parfois c'est le coût, d'autres fois la capacité calorifique ou l'efficacité de l'armement, etc. L'unique condition à laquelle le coefficient d'utilité doit satisfaire est que l'utilité totale du matériel implanté soit définie par la somme des coefficients d'utilité des objets installés. Ainsi, le coût total du matériel mis en place se compose des prix de différents objets, la capacité calorifique totale du carburant transporté est une combinaison linéaire des capacités calorifiques de différents types de combustibles, l'espérance mathématique du nombre de buts touchés

par les types d'armement considérés se présente comme la somme des caractéristiques d'efficacité de chaque type d'armement, etc.

Les n objets ne peuvent pas être tous installés dans des emballages du volume imposé. De plus, la capacité portante admissible des emballages est inférieure à la somme des poids des objets.

Il faut, en respectant les contraintes pondérales et volumiques, assurer une répartition des objets telle que l'utilité totale du matériel mis en place soit maximale.

Souvent, au lieu de contraintes sur le volume, il faut respecter des contraintes sur certaines dimensions. Le nombre de contraintes s'accroît encore si l'emballage et les objets disposés ont une configuration complexe.

Supposons que x_j soit égal à un si le j-ième objet est accepté dans l'emballage et à zéro dans le cas contraire.

Introduisons les notations suivantes:

c_j est le coefficient d'utilité du j-ième objet;

a_{ij}, la i-ème caractéristique du j-ième objet (par exemple, poids, volume ou certaines dimensions);

b_i, la contrainte imposée à la i-ème caractéristique.

Dans les notations adoptées, le problème de l'emballage rationnel du matériel se ramène à maximiser la forme linéaire

$$\sum_{j=1}^{n} c_j x_j \tag{1.2}$$

sous les conditions

$$\sum_{j=1}^{n} a_{ij} x_j \leqslant b_i, \quad i = 1, 2, \ldots, m, \tag{1.3}$$

$$0 \leqslant x_j \leqslant 1, \quad j = 1, 2, \ldots, n, \tag{1.4}$$

où m est le nombre de caractéristiques qui délimitent le choix d'un objet.

Les conditions (1.3)-(1.4) ne sont généralement pas suffisantes pour assurer que les composantes x_j soient égales à zéro ou à l'unité. Aussi doivent-elles être complétées par la condition

$$x_j \text{ entiers pour } j = 1, 2, \ldots, n. \tag{1.5}$$

Le problème (1.2)-(1.5) est un modèle typique de programmation en nombres entiers.

Le cas particulier du problème (1.2)-(1.5) où le nombre m des caractéristiques à prendre en compte dans les contraintes (1.3) est égal à 1 est connu dans la littérature sous le nom de *problème du sac à dos*. Ce problème est l'une des toutes premières formulations qui aient stimulé le développement de la programmation en nombres entiers.

1.3. Par problème extrémal de type combinatoire, nous entendrons toute formulation dans laquelle on demande de faire un choix entre un nombre fini de variantes fournissant la valeur extrémale (maximale ou minimale) d'une fonction économique fixée. On parvient d'habitude à faire correspondre à chaque variante un point de l'espace de dimension s pour un certain s lié à la nature même du problème. Le problème combinatoire se ramène alors à maximiser la fonction économique $F(X)$ définie sur un système fini de points X de l'espace de dimensions.

Les problèmes combinatoires présentant un intérêt pratique ont généralement un nombre de variantes possibles si grand que leur résolution par une révision directe de toutes les variantes comporte d'insurmontables difficultés non seulement pour les ordinateurs modernes, mais encore pour ceux d'un avenir prévisible. C'est pourquoi l'analyse d'un tel problème n'est possible qu'à la condition de créer pour lui une méthode spéciale de résolution permettant de réduire considérablement le nombre de variantes à examiner.

C'est le domaine où dans plusieurs cas l'apport de la programmation linéaire peut être très utile.

Supposons que $F(X)$ soit une fonction linéaire du point X. Soient T l'ensemble des variantes du problème donné ; G, le polyèdre convexe composé de toutes les combinaisons convexes possibles de points T. Il est clair que le problème combinatoire concerné se ramène au problème de programmation linéaire qui consiste à maximiser ou à minimiser la fonction $F(X)$ sur le polyèdre convexe G. Toutefois, pour utiliser les méthodes de programmation linéaire, il faut que le polyèdre G soit donné par un système d'égalités et d'inégalités qui associent entre elles les composantes du point X, c'est-à-dire qu'il soit présenté comme la partie commune d'un certain nombre d'hyperplans et de demi-espaces de l'espace des points X. Si l'on réussit à trouver la représentation équivalente du polyèdre G, le problème est alors accessible aux méthodes de la programmation linéaire. A titre d'exemple, on peut se référer au problème d'affectation (cf. [87], § 4 du chapitre 14), dans lequel il est facile de former le système de contraintes linéaires donnant la description du polyèdre correspondant. Malheureusement, les tentatives de construire le système le plus économique de conditions de définition du polyèdre G ou un système qui s'en approche n'ont pas été jusqu'à présent couronnées de succès pour beaucoup de problèmes combinatoires importants, bien qu'en principe la possibilité d'une telle construction ne soulève aucun doute (cf. [87], théorème 3.1 du chapitre 5). Toutefois, il n'y a qu'à permettre d'utiliser pour la donnée de G, outre les conditions linéaires, des contraintes d'intégralité du type (1.1) pour faire disparaître généralement les difficultés relevées.

Actuellement, les problèmes combinatoires connus sont dans leur majorité ramenés à des problèmes de programmation linéaire en nombres entiers.

Considérons le *problème du voyageur de commerce* qui fait l'objet d'une littérature abondante. Jusqu'à des temps récents, il n'y avait pas d'algorithme valable pour la résolution de ce problème. La situation a quelque peu changé avec la publication de [47]. Dans cet ouvrage on propose une méthode qui permet, d'après ce qu'affirment les auteurs, de résoudre sur des calculateurs digitaux puissants les problèmes du voyageur de commerce à quelques dizaines de points.

Enonçons ce problème en termes qui ont défini sa dénomination, bien qu'il en existe d'autres applications présentant un intérêt pratique bien plus grand. Soient n points $p_1, p_2, \ldots, p_n$. Supposons que le trajet du point p_i au point p_j nécessite t_{ij} unités de temps. La tâche du voyageur de commerce qui se trouve en l'un des points p_i est de faire la tournée de tous les points en passant par chacun d'eux juste une fois et revenir au point de départ après avoir effectué tout le trajet en un temps le plus petit possible. Nous avons là un problème extrémal combinatoire typique, le nombre de ses variantes étant $(n-1)!$. Chaque variante peut être identifiée à un système de nombres x_{ij}, $i, j = 1, 2, \ldots, n$; $i \neq j$, en égalant x_{ij} à 1 ou à 0 suivant que le voyageur de commerce quitte ou non le point p_i pour le point p_j. Le temps total nécessaire pour la réalisation de la variante $X = \{x_{ij}\}$ est évidemment

$$t(X) = \sum_{i=1}^{n} \sum_{j=1\,(j\neq i)}^{n} t_{ij} x_{ij}. \tag{1.6}$$

Marquons sur le plan n points associés aux points p_i. Chaque variante X du parcours peut être représentée graphiquement par un itinéraire orienté fermé γ_X comprenant tous les n points (fig. 5.1, *a*).

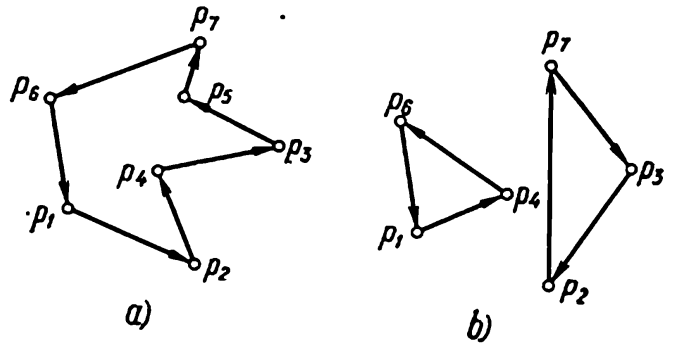

Fig. 5.1

La relation entre les formes graphique et analytique de la variante de la tournée est évidente: x_{ij} est égal à 0 ou à 1 suivant que la

communication $\overrightarrow{p_i p_j}$ entre dans γ_x ou non. Procédons à la formation des conditions qui délimitent le choix des systèmes de nombres $\{x_{ij}\}$ (cf. [55]). Avant tout, nous tiendrons compte du fait que de chaque point p_i il sort juste une communication de l'itinéraire fermé γ_x, et que dans chaque point p_j n'entre également qu'une communication de cet itinéraire.

Ces circonstances sont équivalentes aux conditions

$$\sum_{j=1(j\neq i)}^{n} x_{ij}=1, \quad i=1, 2, \ldots, n, \tag{1.7}$$

$$\sum_{i=1(i\neq j)}^{n} x_{ij}=1, \quad j=1, 2, \ldots, n. \tag{1.8}$$

Par définition, chacun des paramètres x_{ij} doit être égal soit à zéro, soit à l'unité. Au lieu de cela, nous exigerons que soient respectées les inégalités

$$0 \leqslant x_{ij} \leqslant 1. \tag{1.9}$$

On sait qu'un sommet quelconque du polyèdre défini par les conditions (1.7)-(1.9) est entier (c'est-à-dire que toutes ses composantes sont des nombres entiers). De plus, chaque itinéraire fermé orienté passant par chacun des n points correspond à un certain sommet du polyèdre (1.7)-(1.9).

Si chaque sommet de ce polyèdre engendrait un itinéraire fermé orienté comprenant les n points, le système de contraintes recherché serait construit et le problème du voyageur de commerce se ramènerait à un problème ordinaire d'affectation. Or, ce n'est pas le cas. Ainsi le système de nombres $\{x_{ij}\}$ correspondant aux deux itinéraires fermés, représentés fig. 5.1, *b*, constitue également un sommet du polyèdre (1.7)-(1.9). Aussi est-il nécessaire de faire appel à des conditions supplémentaires qui rendent impossible la division des itinéraires fermés de longueur n en plusieurs itinéraires moins longs. Associons au point p_i un nombre inconnu u_i avec $i = 1, 2, \ldots, n-1$, et justifions les relations

$$u_j - u_i + (n-1)\, x_{ij} \leqslant n-2,$$
$$i, j = 1, 2, \ldots, n-1, \; i \neq j. \tag{1.10}$$

Supposons que $\{x_{ij}\}$ corresponde à plusieurs itinéraires fermés. Montrons qu'il n'existe pas alors de nombres u_i qui avec $\{x_{ij}\}$ satisfassent aux conditions (1.10). Considérons en effet l'un des itinéraires fermés correspondant à $\{x_{ij}\}$ qui ne passe pas par p_n (un tel itinéraire existe d'avance, car autrement $\{x_{ij}\}$ correspondrait à un itinéraire fermé unique). La sommation des inégalités (1.10) associées à toutes les communications de cet itinéraire conduit à la relation impossible

$$n-1 \leqslant n-2.$$

D'autre part, si $x_{ij}\}$ correspond à seul itinéraire fermé γ_x passant par tous les points p_i, les conditions (1.10) sont alors respectées. En effet, supposons que γ_x passe successivement par $p_{i_1}, p_{i_2}, \ldots$ $\ldots, p_{i_{n-1}}, p_{i_n} = p_n$. Posons

$$u_{i_s} = -(s-1) \text{ pour } s = 1, 2, \ldots, n-1.$$

On a :

$$\text{a) } u_{i_s} - u_{i_{s-1}} + (n-1)\, x_{i_{s-1} i_s} = -1 + (n-1) = n-2$$
$$\text{avec } s = 1, \ldots, n-1\,;$$
$$\text{b) } u_i - u_j + (n-1)\, x_{ij} = u_i - u_j \leqslant 0 - (-n+2) = n-2$$
$$\text{avec } \overrightarrow{p_i p_j} \notin \gamma_x, \quad i, j \neq n.$$

Ainsi, pour qu'un système de paramètres $\{x_{ij}\}$ vérifiant les contraintes (1.7)-(1.10) corresponde à un seul itinéraire fermé, il faut et il suffit d'imposer à ces paramètres d'être entiers :

$$x_{ij} \text{ entiers pour tout } i \text{ et } j. \tag{1.11}$$

Ainsi qu'on l'a montré, un système de paramètres $\{x_{ij}\}$ correspondant à un seul itinéraire fermé rend possible le choix de nombres entiers u_i associés à $\{x_{ij}\}$ par les conditions (1.10). C'est pourquoi la contrainte d'intégralité peut être étendue également aux inconnues u_i, $i = 1, 2, \ldots, n-1$.

Le problème du voyageur de commerce est ainsi ramené au problème de programmation linéaire qui consiste à minimiser la forme linéaire (1.6) sous les conditions (1.7)-(1.10) et sous l'hypothèse que toutes les inconnues soient entières.

Mentionnons encore un problème extrémal combinatoire d'une grande portée pratique, dit problème général de la théorie des horaires. Voici l'une de ses formulations.

Soient m machines-outils et n pièces. Chaque pièce doit être usinée sur plusieurs machines ; de plus, au cours de l'usinage, les machines doivent être utilisées dans un ordre déterminé. On connaît le temps t_{ij} nécessaire pour le façonnage de la j-ième pièce sur la i-ème machine. On connaît également le temps s_{ijk} de réglage de la i-ème machine nécessaire pour passer de l'usinage de la j-ième à celui de la k-ième pièce. L'ordre de travail à établir pour chaque machine doit être tel que le temps total absorbé par l'usinage de toutes les pièces soit minimal. On dispose actuellement de méthodes exactes de résolution valables seulement pour certains cas particuliers du problème de la théorie des horaires. Plusieurs auteurs (cf., par exemple, [7, 12, 52]) ramènent le problème général de la théorie des horaires à la programmation en nombres entiers.

1.4. Il existe actuellement de nombreuses méthodes pour résoudre les problèmes extrémaux non linéaires. Cependant, une grande

partie de ces méthodes est conçue pour la discussion de problèmes dits convexes, dans lesquels la fonction économique à maximiser (minimiser) est convexe vers le haut (vers le bas) et l'ensemble découpé par le système de contraintes est convexe. Cette situation est une conséquence de ce que tout optimum local d'un problème de programmation convexe en est son optimum global.

Les méthodes de résolution des problèmes extrémaux qui dépassent le cadre de la programmation convexe sont très faiblement développées. C'est pourquoi la possibilité de ramener à des problèmes de programmation en nombres entiers une série de problèmes importants de ce type présente un grand avenir.

Considérons le *problème* dit *de distribution* dont la formulation la plus simple est donnée ci-dessous.

Supposons qu'il existe m centres de production $A_1, A_2, \ldots, A_m$ d'un certain produit homogène. Les usagers de ce produit sont les points $B_1, B_2, \ldots, B_n$, dans lesquels la demande respective est $b_1, b_2, \ldots, b_n$ unités de produit. On connaît les coûts c_{ij} de transport d'une unité de produit de A_i à B_j. En outre, pour chaque point de production A_i, il existe une relation entre le prix de revient d'une unité de produit (compte tenu des investissements de capitaux spécifiques) et le volume de fabrication définie par la fonction

$$\varphi_i(X_i),$$

où X_i est le volume de production en A_i.

Si l'on désigne par x_{ij} la quantité de transport de A_i à B_j, le coût total de production et de transport du produit peut alors s'écrire

$$F(X) = \sum_{i=1}^{m} \sum_{j=1}^{n} \left[c_{ij} + \varphi_i\left(\sum_{j=1}^{n} x_{ij}\right)\right] x_{ij}. \tag{1.12}$$

Les contraintes

$$\sum_{i=1}^{m} x_{ij} = b_j, \quad j = 1, 2, \ldots, n, \tag{1.13}$$

$$x_{ij} \geqslant 0, \quad i = 1, 2, \ldots, m; \quad j = 1, 2, \ldots, n, \tag{1.14}$$

assurent la satisfaction de tous les usagers et l'unicité de la direction dans laquelle s'effectue chaque transport.

Le problème de distribution consiste à minimiser la fonction (1.12) sous les conditions (1.13)-(1.14).

Dans la plupart des cas, la structure des coûts de production est telle que (1.12) est une fonction convexe vers le haut (avec la croissance de la production diminuent les charges imposées par son augmentation d'une unité). C'est pourquoi (1.12)-(1.14) n'est pas un problème de programmation convexe. Dans les problèmes de distribution pratiques, il arrive souvent que le volume de production recherché soit limité par les normes en vigueur et puisse prendre,

par conséquent, l'une de quelques valeurs imposées d'avance. Examinons ce cas précis.

Supposons que les valeurs éventuelles du volume de production en A_i soient $d_{\lambda i}$, $\lambda = 0, 1, \ldots, s_i$. Convenons que

$$d_{0i} = 0 < d_{1i} < \ldots < d_{s_i i}.$$

Introduisons, en plus des variables x_{ij}, des variables additionnelles $y_{\lambda i}$, où i correspond au point A_i et λ ($\lambda = 1, 2, \ldots, s_i$), au volume $d_{\lambda i}$ de la production en ce point.

Exprimons la quantité X_i de la production en A_i à l'aide des variables $y_{\lambda i}$:

$$X_i = \sum_{\lambda=1}^{s_i} (d_{\lambda i} - d_{\lambda-1,\, i})\, y_{\lambda i}. \tag{1.15}$$

Pour égaliser X_i à l'une des valeurs possibles $d_{\lambda i}$, le système de nombres $y_{\lambda i}$ doit jouir de la propriété

$$\begin{gathered} y_{1i} = y_{2i} = \ldots = y_{\lambda_0 i} = 1\,; \\ y_{\lambda_0+1,\, i} = y_{\lambda_0+2,\, i} = \ldots = y_{s_i i} = 0\,; \\ 0 \leqslant \lambda_0 \leqslant s_i. \end{gathered} \tag{1.16}$$

Les conditions (1.16) sont équivalentes aux contraintes

$$\left.\begin{aligned} y_{\lambda-1,\, i} &\geqslant y_{\lambda i}, \quad \lambda = 2, \ldots, s_i, \\ 0 \leqslant y_{\lambda i} &\leqslant 1, \quad \lambda = 1, 2, \ldots, s_i, \end{aligned}\right\} \tag{1.17}$$

et à la restriction d'intégralité des variables $y_{\lambda i}$. Désignons le coût de production du volume $d_{\lambda i}$ en A_i par

$$\psi_{\lambda i} = d_{\lambda i} \varphi_i\, (d_{\lambda i}).$$

Les frais de production en A_i peuvent être également exprimés à l'aide des variables $y_{\lambda i}$:

$$X_i \varphi_i\, (X_i) = \sum_{\lambda=1}^{s_i} (\psi_{\lambda i} - \psi_{\lambda-1,\, i})\, y_{\lambda i}, \tag{1.18}$$

où $y_{\lambda i}$ sont liées par les conditions (1.17) et les restrictions d'intégralité.

Les relations (1.15) et (1.18) permettent d'énoncer le problème de distribution sous la forme équivalente suivante. Minimiser la forme linéaire

$$\sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} + \sum_{i=1}^{m} \sum_{\lambda=1}^{s_i} (\psi_{\lambda i} - \psi_{\lambda-1,\, i})\, y_{\lambda i} \tag{1.19}$$

sous les conditions

$$\sum_{i=1}^{m} x_{ij} = b_j, \quad j = 1, 2, \ldots, n, \tag{1.20}$$

$$\sum_{j=1}^{n} x_{ij} \leqslant \sum_{\lambda=1}^{s_i} (d_{\lambda i} - d_{\lambda-1, i}) y_{\lambda i}, \quad i = 1, 2, \ldots, m, \tag{1.21}$$

$$x_{ij} \geqslant 0, \quad i = 1, 2, \ldots, m; \quad j = 1, 2, \ldots, n, \tag{1.22}$$

$$0 \leqslant y_{\lambda i} \leqslant 1, \quad \lambda = 1, 2, \ldots, s_i; \quad i = 1, 2, \ldots, m, \tag{1.23}$$

$$y_{\lambda-1, i} \geqslant y_{\lambda i}, \quad \lambda = 2, 3, \ldots, s_i; \quad i = 1, 2, \ldots, m, \tag{1.24}$$

$$y_{\lambda i} \text{ entiers}, \ \lambda = 1, 2, \ldots, s_i; \quad i = 1, 2, \ldots, m. \tag{1.25}$$

Si les paramètres b_j et $d_{\lambda i}$ sont des entiers (dans les problèmes pratiques, on peut toujours les rendre tels en choisissant une unité de mesure appropriée), alors, pour des $y_{\lambda i}$ entiers, les transports x_{ij} peuvent aussi être considérés comme des entiers.

Ainsi, le problème de distribution non linéaire se ramène à un problème de programmation linéaire en nombres entiers.

Supposons maintenant que le volume de production X_i en A_i puisse prendre une valeur arbitraire délimitée par le segment $[0, a_i]$. Approximons la fonction $\psi_i(X) = X\varphi_i(X)$ à l'aide d'une fonction linéaire par morceaux $\widetilde{\psi}_i(X)$ aux segments de linéarité $[d_{\lambda i}, d_{\lambda+1, i}]$, $\lambda = 0, 1, \ldots, s_{i-1}$; $(d_{0i} = 0;\ d_{s_i i} = a_i)$. Plus le nombre s_i est grand, plus $\widetilde{\psi}_i(X)$ est proche de $\psi_i(X)$. Remplaçons la fonction économique (1.12) du problème (1.12)-(1.14) par sa valeur approchée

$$\widetilde{F}(X) = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij} + \sum_{i=1}^{m} \widetilde{\psi}_i \Big(\sum_{j=1}^{n} x_{ij}\Big). \tag{1.26}$$

Introduisons, comme auparavant, les variables $y_{\lambda i}$ à l'aide desquelles nous donnerons, d'après les formules (1.15) et (1.18), l'expression de X_i et de $\widetilde{\psi}_i(X_i) = X_i \widetilde{\varphi}_i(X_i)$. Pour que le point $(X_i, \widetilde{\psi})$ appartienne à la ligne brisée

$$Y_i = \widetilde{\psi}_i(X_i),$$

les variables $y_{\lambda i}$ doivent être limitées par les conditions

$$\begin{gathered} y_{1i} = y_{2i} = \ldots = y_{\lambda_0-1, i} = 1; \\ y_{\lambda_0+1, i} = y_{\lambda_0+2, i} = \ldots = y_{s_i i} = 0; \\ 0 \leqslant y_{\lambda_0 i} \leqslant 1, \end{gathered} \tag{1.27}$$

où λ_0 est un entier quelconque compris entre 1 et s_i.

On vérifie aisément (cf. exercice 1) que (1.27) est équivalente aux restrictions

$$y_{\lambda-1,i} \geqslant \delta_{\lambda i} \geqslant y_{\lambda i}, \quad \lambda = 2, 3, \ldots, s_i, \tag{1.28}$$

$$0 \leqslant y_{\lambda i} \leqslant 1; \tag{1.29}$$

$$\delta_{\lambda i} \text{ entiers.} \tag{1.30}$$

Par conséquent, le problème (1.26), (1.13), (1.14) se ramène à la minimisation de la forme linéaire (1.19) sous les conditions (1.20), (1.21) (où les inégalités sont remplacées par des égalités), (1.22) et (1.23), et tout en respectant les contraintes (1.28) et (1.30) pour tout i. Nous avons obtenu un problème de programmation partiellement en nombres entiers.

Le procédé utilisé ici lors de la construction du problème en nombres entiers équivalent peut être appliqué à d'autres modèles aux éléments non linéaires.

1.5. Dans plusieurs cas, l'introduction de variables astreintes à être entières fournit la possibilité de se donner des ensembles non convexes au moyen de systèmes de contraintes linéaires.

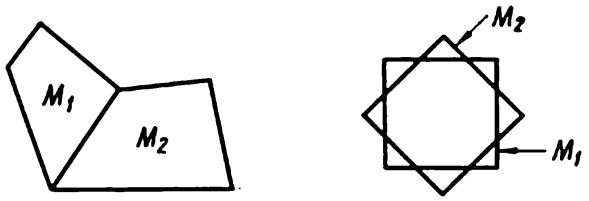

Fig. 5.2

Soit, par exemple, l'ensemble M, réunion de deux polyèdres convexes M_1 et M_2 (fig. 5.2), dont les systèmes de contraintes s'écrivent

$$L_i'(X) = \sum_{j=1}^{n} a_{ij}' x_j \leqslant b_i', \; i = 1, 2, \ldots, m_1, \tag{1.31}$$

et

$$L_i''(X) = \sum_{j=1}^{n} a_{ij}'' x_j \leqslant b_i'', \; i = 1, 2, \ldots, m_2. \tag{1.32}$$

Supposons que l'on connaisse un nombre b tel qu'il vérifie les inégalités

$$\left.\begin{aligned} &\max_{X \in M} L_i'(X) \leqslant b_i' + b, \; i = 1, \ldots, m_1, \\ &\max_{X \in M} L_i''(X) \leqslant b_i'' + b, \; i = 1, \ldots, m_2. \end{aligned}\right\} \tag{1.33}$$

Le polyèdre M est alors défini par le système de contraintes suivant :

$$L_i'(X) \leqslant b_i' + b\delta, \quad i = 1, 2, \ldots, m_1, \tag{1.34}$$

$$L_i''(X) \leqslant b_i'' + b(1 - \delta), \quad i = 1, 2, \ldots, m_2, \tag{1.35}$$

$$0 \leqslant \delta \leqslant 1, \tag{1.36}$$

$$\delta \text{ entier.} \tag{1.37}$$

En effet, si $X \in M$, alors $X \in M_1$, ou bien $X \in M_2$. Posons, pour fixer les idées, $X \in M_1$. Dans ce cas, $(X, \delta) = (X, 0)$ vérifie les conditions (1.34)-(1.37), puisque (1.35) est corollaire de (1.33). Supposons d'autre part que (X, δ) obéisse aux conditions (1.34)-(1.37). En vertu de (1.36), (1.37), δ est égal soit à 0, soit à 1. Si $\delta = 0$, d'après (1.34), le point $X \in M_1$. Mais si $\delta = 1$, conformément à (1.35), le point $X \in M_2$. Par conséquent, $X \in M$.

On peut également recourir au procédé décrit en présence d'un plus grand nombre de polyèdres convexes composant le polyèdre M (cf. exercice 5).

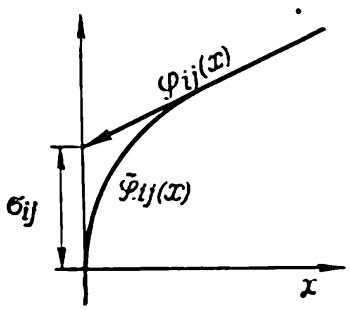

Fig. 5.3

Ainsi, le problème de l'optimisation d'une fonction linéaire définie sur un ensemble composé de plusieurs polyèdres convexes (ensemble qui, en règle générale, n'est pas convexe) se ramène en principe à un problème de programmation linéaire partiellement en nombres entiers.

A titre de conclusion à ce paragraphe, voici un exemple de problème non linéaire à fonction économique discontinue, facilement réductible à un problème de programmation en nombres entiers. Il s'agit de la constitution du programme optimal de transport d'un produit homogène entre des points de production A_i aux volumes de fabrication a_i $(i = 1, 2, \ldots, m)$ et de points de consommation B_j aux volumes de consommation b_j $(j = 1, 2, \ldots, n)$. Mais à la différence du problème de transport classique, le coût de transport $\varphi_{ij}(x_{ij})$ de la livraison x_{ij} est de la forme

$$\varphi_{ij}(x_{ij}) = \begin{cases} c_{ij}x_{ij} + \sigma_{ij}, & x_{ij} > 0 \quad (\sigma_{ij} > 0); \\ 0, & x_{ij} = 0. \end{cases} \tag{1.38}$$

La structure des dépenses de transport (1.38) correspond, par exemple, à celle du tarif des taxis, selon lequel le prix de la course se compose d'une somme fixe (10 kopecks) servant à acquitter la location de la voiture et d'une partie proportionnelle à la distance (dans le problème considéré, le rôle de la distance est joué par la valeur du transport). La fonction (1.38), représentée figure 5.3,

est discontinue en $x = 0$. On pourrait certes remplacer $\varphi_{ij}(x)$ par la fonction continue voisine $\tilde{\varphi}_{ij}(x)$ qui est également portée sur la figure 5.3. Mais une telle approximation ne simplifie rien du fait que $\tilde{\varphi}_{ij}(x)$ est une fonction convexe vers le haut et que le problème de minimisation obtenu ne cadre pas avec le schéma de programmation convexe.

Pour ramener le problème énoncé à un problème de programmation en nombres entiers, servons-nous d'un procédé qui s'est avéré utile lors de l'analyse de domaines non convexes. Le coût de transport total peut alors être mis sous la forme

$$\sum_{i=1}^{m} \sum_{j=1}^{n} (c_{ij}x_{ij} + \sigma_{ij}y_{ij}), \tag{1.39}$$

si l'on exige de plus que

$$y_{ij} = \begin{cases} 0 & \text{avec} \quad x_{ij} = 0, \\ 1 & \text{avec} \quad x_{ij} > 0. \end{cases} \tag{1.40}$$

$$(i = 1, 2, \ldots, m; \quad j = 1, 2, \ldots, n).$$

La quantité x_{ij} ne peut évidemment pas dépasser le nombre

$$d_{ij} = \min (a_i, b_j).$$

C'est pourquoi les conditions (1.40) sont équivalentes aux contraintes

$$\left.\begin{array}{ll} x_{ij} \leqslant d_{ij}y_{ij}, & \\ & (i = 1, \ldots, m; \quad j = 1, \ldots, n) \\ y_{ij} \text{ entier.} & \end{array}\right\} \tag{1.41}$$

En effet, pour $x_{ij} > 0$, la variable y_{ij} doit être au moins égale à 1. Comme l'expression (1.39) doit être minimisée, alors que $x_{ij} \leqslant d_{ij}$, on a $y_{ij} = 1$. Si par contre $x_{ij} = 0$, s'efforcer de minimiser (1.39) conduit à prendre $y_{ij} = 0$. Ainsi, en présence du coût indépendant de la quantité de transport, mais lié à l'existence même de ce transport, le problème de transport consiste à minimiser la forme linéaire (1.39) sous les conditions (1.41) et les contraintes de transport usuelles

$$\sum_{j=1}^{n} x_{ij} = a_i, \quad i = 1, 2, \ldots, m, \tag{1.42}$$

$$\sum_{i=1}^{m} x_{ij} = b_j, \quad j = 1, 2, \ldots, n, \tag{1.43}$$

$$x_{ij} \geqslant 0, \quad i = 1, 2, \ldots, m; \quad j = 1, 2, \ldots, n. \tag{1.44}$$

Les programmes de base du problème de transport étant astreints être entiers (tous les nombres a_i et b_j sont supposés entiers), on

en tire que (1.39), (1.41)-(1.44) est un problème de programmation linéaire en nombres entiers.

Les exemples de problèmes importants et complexes cités dans ce paragraphe sont loin d'épuiser toutes les applications possibles de la programmation en nombres entiers. Toutefois, ce qui a été rapporté ici suffit amplement pour se rendre compte des possibilités énormes que recèle cette branche de la programmation.

Dans [37] on peut trouver certaines applications intéressantes non exposées dans ce paragraphe.

Il peut sembler à première vue que la voie naturelle de l'exploration d'un problème en nombres entiers consiste à résoudre le problème linéaire correspondant en arrondissant ensuite les composantes de son programme optimal aux entiers les plus proches. Or, dans la plupart des cas, une telle voie non seulement éloigne de l'optimum, mais ne donne même pas un programme (il en est ainsi notamment dans les exemples numériques des §§ 3, 4). Les problèmes de programmation en nombres entiers nécessitent des méthodes d'analyse spéciales.

L'essentiel de ce chapitre est consacré à l'exposé des méthodes de programmation en nombres entiers qui sont les mieux étudiées à l'heure actuelle.

§ 2. Conditions d'intégralité des ensembles convexes

2.1. Considérons le problème de programmation linéaire qui consiste à maximiser une certaine forme linéaire des variables x_j sous les conditions

$$\sum_{j=1}^{n} a_{ij}x_j = b_i, \quad i = 1, 2, \ldots, m, \tag{2.1}$$

$$x_j \geqslant 0, \quad j = 1, 2, \ldots, n, \tag{2.2}$$

qui définissent l'ensemble convexe M. Le rang de la matrice $A = \| a_{ij} \|$ est supposé égal à m. Quel est le cas succeptible d'assurer l'intégralité de toutes les composantes de la solution du problème?

Autrement dit, il faut trouver les conditions qui rendraient inutile la justification de l'hypothèse supplémentaire sur les variables x_j astreintes à être entières et ramèneraient par conséquent la procédure à l'analyse d'un problème de programmation linéaire ordinaire.

Nous dirons que le programme d'un problème de programmation linéaire est *entier* si toutes ses composantes sont des nombres entiers. Si la forme linéaire du problème n'est pas fixée, l'optimum peut être atteint sur n'importe quel sommet de l'ensemble M. Par conséquent, pour que les solutions de base des problèmes à forme linéaire

arbitraire et à ensemble convexe des conditions M soient entières, il faut et il suffit que toutes les composantes de n'importe quel sommet de M soient des nombres entiers, c'est-à-dire que tous les sommets de M soient *entiers*. Un polyèdre ou un ensemble convexe qui jouit de la propriété indiquée est dit entier. L'exemple d'un tel ensemble est donné par le polyèdre des conditions du problème de transport.

Le présent paragraphe indique quelques conditions qui assurent l'intégralité des ensembles convexes. On expose ici, sous une forme légèrement modifiée, certains résultats obtenus par [31] et [76].

2.2. L'ensemble convexe M dépend de tous les paramètres qui font partie des conditions (2.1). Dans ce qui suit, cependant, les éléments de la matrice A seront considérés comme des entiers fixes, et les composantes b_i du vecteur des contraintes B comme des entiers quelconques. C'est pourquoi, dans la désignation de l'ensemble (2.1)-(2.2), il est rationnel de traduire cette dépendance du seul vecteur B en introduisant pour M la nouvelle notation $M(B)$.

Nous allons énoncer et démontrer une proposition qui met en évidence une classe de matrices A telles que tout ensemble convexe du type $M(B)$ soit astreint à être entier.

Théorème 2.1. *Pour imposer la condition d'intégralité à tous les ensembles $M(B)$, où B est un vecteur arbitraire de dimension m à composantes entières, il faut et il suffit que tout mineur d'ordre m de la matrice $A = \| a_{ij} \|_{m.n}$ soit égal à 0 ou à ± 1.*

Condition suffisante. Soit la matrice A qui jouit de la propriété indiquée et $X = (x_1, x_2, \ldots, x_n)$ le sommet arbitraire de l'ensemble $M(B)$, c'est-à-dire le programme de base associé aux conditions (2.1)-(2.2), où B est un vecteur arbitraire de composantes entières. Si les vecteurs $A_{i_1}, A_{i_2}, \ldots, A_{i_m}$ constituent la base du programme X, alors $x_j = 0$ avec $j \neq i_\lambda$ ($\lambda = 1, 2, \ldots, m$);

$$\sum_{\lambda=1}^{m} x_{i_\lambda} A_{i_\lambda} = B. \tag{2.3}$$

C'est pourquoi la démonstration du fait que le sommet X est astreint à être entier se ramène à établir que toutes les composantes x_{i_λ} du système d'équations (2.3) sont entières.

Posons

$$A_X = (A_{i_1}, A_{i_2}, \ldots, A_{i_m}); \quad \hat{X} = (x_{i_1}, x_{i_2}, \ldots, x_{i_m})^T.$$

Dans les nouvelles notations, le système (2.3) s'écrit

$$A_X \hat{X} = B.$$

Par conséquent

$$\hat{X} = A_X^{-1} B. \tag{2.4}$$

Les définitions de la matrice inverse et de l'intégralité des éléments de la matrice A_X entraînent que chaque élément de la matrice A_X^{-1} est une fraction rationnelle dont le dénominateur est le déterminant $|A_X|$. Par condition, $|A_X| = \pm 1$. Donc, tous les éléments de la matrice A_X^{-1} sont des nombres entiers. En tenant compte de ce que les composantes du vecteur B sont des nombres entiers et de la formule (2.4), on obtient que toutes les composantes du vecteur $\hat{X}$ sont des nombres entiers.

Condition nécessaire. Supposons que les sommets des ensembles convexes $M(B)$ soient entiers quel que soit B, mais qu'il existe pourtant un mineur

$$\Delta = |(A_{i_1}, A_{i_2}, \ldots, A_{i_m})|$$

de la matrice A qui soit différent de 0 ou de ± 1.

Les éléments de la matrice A étant astreints à être entiers, il en résulte l'inégalité

$$|\Delta| \geqslant 2.$$

Donc, le déterminant de la matrice $(A_{i_1}, A_{i_2}, \ldots, A_{i_m})^{-1}$, égal à $\frac{1}{\Delta}$, ne peut pas être un nombre entier. Ceci, à son tour, fait qu'il est impossible que tous les éléments de la matrice $(A_{i_1}, A_{i_2}, \ldots, A_{i_m})^{-1}$ soient entiers. Supposons que la t-ième colonne de la matrice $(A_{i_1}, A_{i_2}, \ldots, A_{i_m})^{-1}$ contienne un élément fractionnaire. Dans ce cas, le vecteur

$$\hat{X}' = (x'_{i_1}, x'_{i_2}, \ldots, x'_{i_m})^T = (A_{i_1}, A_{i_2}, \ldots, A_{i_m})^{-1} e_t \tag{2.5}$$

comporte des composantes fractionnaires (e_t est le vecteur unité avec l'unité à la t-ième place). Cherchons un entier α tel que toutes les composantes du vecteur

$$\hat{X} = (x_{i_1}, x_{i_2}, \ldots, x_{i_m})^T = (x'_{i_1} + \alpha, x'_{i_2} + \alpha, \ldots, x'_{i_m} + \alpha)^T$$

soient positives.

Il est clair que

$$(A_{i_1}, A_{i_2}, \ldots, A_{i_m})\hat{X} = e_t + \alpha \sum_{\lambda=1}^{m} A_{i_\lambda},$$

c'est-à-dire que le point $X = (x_1, x_2, \ldots, x_n)$ aux composantes

$$x_j = \begin{cases} 0 & \text{avec } j \neq i_\lambda \\ x_{i_\lambda} & \text{avec } j = i_\lambda, \end{cases} \quad \lambda = 1, 2, \ldots, m,$$

constitue un sommet du polyèdre $M(B)$ avec

$$B = e_t + \alpha \sum_{\lambda=1}^{m} A_{i_\lambda}.$$

Dans la mesure où le vecteur $\hat{X}'$ compte des composantes fractionnaires et où α est un entier, il y a parmi les nombres x_{i_λ} des nombres fractionnaires, ce qui signifie que le sommet X n'est pas entier. Cette réduction à l'absurde justifie les conditions du théorème.

2.3. Supposons maintenant que la matrice A contienne les vecteurs-colonnes unités $e_1, e_2, \ldots, e_m$, c'est-à-dire qu'elle soit de la forme

$$A = (\overline{A}, e_1, e_2, \ldots, e_m).$$

Soit Δ_m un mineur quelconque d'ordre m de la matrice A contenant des colonnes unités. En développant le déterminant Δ_m par rapport aux colonnes unités qui le composent, on obtient

$$\Delta_m = \pm\overline{\Delta}_s, \tag{2.6}$$

où $\overline{\Delta}_s$ est le mineur d'ordre $s < m$ de la matrice $\overline{A}$.

D'autre part, si Δ_s est un mineur quelconque d'ordre $1 \leqslant s < m$ de la matrice $\overline{A}$ composé d'éléments des lignes $i_1, i_2, \ldots, i_s$ et des colonnes $j_1, j_2, \ldots, j_s$ de cette matrice, on a alors pour le mineur Δ_m d'ordre m de la matrice A du type

$$\Delta_m = |(A_{j_1}, A_{j_2}, \ldots, A_{j_s}, e_{k_1}, e_{k_2}, \ldots, e_{k_{m-s}})|$$

l'égalité (2.6). On entend ici par $k_1, k_2, \ldots, k_{m-s}$ le système d'indices formé à partir de $(1, 2, \ldots, m)$ par élimination des numéros $i_1, i_2, \ldots, i_s$.

Des relations obtenues entre les mineurs des matrices A et $\overline{A}$ il résulte que pour égaliser tous les mineurs d'ordre m à 0 ou à ± 1, il faut et il suffit que tous les mineurs de la matrice $\overline{A}$ soient égaux à 0 ou à ± 1.

Supposons que le système des conditions définissant le polyèdre convexe $M_1(B)$ s'écrit

$$\sum_{j=1}^{n} a_{ij}x_j \leqslant b_i, \quad i = 1, 2, \ldots, m, \tag{2.7}$$

$$x_j \geqslant 0, \quad j = 1, 2, \ldots, n. \tag{2.8}$$

Les considérations précédentes relatives aux mineurs des matrices A et $\overline{A}$ ainsi que le théorème 2.1 justifient la proposition suivante.

Théorème 2.2. *Pour que tous les ensembles convexes $M_1(B)$ définis par les conditions* (2.7)-(2.8), *où B est un vecteur entier arbitraire, soient entiers, il faut et il suffit que tous les mineurs de la matrice A soient égaux à* 0 *ou à* ± 1.

Considérons les ensembles convexes $M(B, \alpha, \beta)$ et $M(B_1, B_2, \alpha, \beta)$ définis respectivement par les relations

$$\sum_{j=1}^{n} a_{ij}x_j = b_i, \quad i = 1, 2, \ldots, m, \tag{2.9}$$

$$\alpha_j \leqslant x_j \leqslant \beta_j, \quad j = 1, 2, \ldots, n, \tag{2.10}$$

et

$$b_i^{(1)} \leqslant \sum_{j=1}^{n} a_{ij}x_j \leqslant b_i^{(2)}, \quad i = 1, 2, \ldots, m, \tag{2.11}$$

$$\alpha_j \leqslant x_j \leqslant \beta_j, \quad j = 1, 2, \ldots, n. \tag{2.12}$$

En utilisant les théorèmes 2.1 et 2.2, on obtient aisément le résultat suivant.

Théorème 2.3. *L'intégralité des ensembles convexes $M(B, \alpha, \beta)$ et $M(B_1, B_2, \alpha, \beta)$ avec des vecteurs entiers arbitraires B, B_1, B_2, α, β résulte de l'intégralité des familles d'ensembles $M(B)$ et $M_1(B)$ respectivement, où les composantes du vecteur B sont des entiers quelconques.*

La démonstration du théorème 2.3 est laissée au lecteur (cf. exercice 6).

2.4. Considérons le problème de programmation linéaire qui consiste à maximiser la forme linéaire définie par le vecteur C sous les conditions (2.7), (2.8). Désignons-le par $L(C, A, B)$. D'après le théorème 2.2, on ne peut garantir l'intégralité des solutions de base des problèmes $L(C, A, B)$ avec C quelconques et B entiers que si tous les mineurs de la matrice A sont égaux à 0 ou à ± 1. C'est là, notamment, le témoignage d'une pauvreté extrême de la classe des matrices A qui engendrent les problèmes aux solutions entières.

Dans ce qui suit, on met en évidence l'une des plus importantes familles de problèmes de programmation linéaire jouissant de la propriété d'intégralité. Cette famille, dont fait partie le problème de transport, compte également des problèmes qui ne sont pas du type « transport ». Considérons l'ensemble Γ des matrices A jouissant des propriétés suivantes:

1. Chaque colonne de A compte pas plus de deux éléments non nuls.
2. Chaque élément non nul de A est égal à 1 ou à -1.
3. Les lignes de la matrice A peuvent être décomposées en deux ensembles disjoints tels que les éléments non nuls de chaque colonne appartiennent à des lignes d'ensembles différents si ces éléments sont égaux entre eux, et se situent dans des lignes du même ensemble si l'un d'eux est 1 et l'autre -1.

Un exemple de matrice appartenant à l'ensemble Γ est donné par la matrice des conditions d'un problème de transport dont le nombre de points est égal à m. Les ensembles des lignes déterminés par la définition de Γ (propriété 3) correspondent dans le cas considéré aux ensembles des points de production et des points de consommation.

Théorème 2.4. *Si la matrice $A \in \Gamma$, avec des C quelconques et des B entiers tels que le problème L (C, A, B) est résoluble, ce dernier comporte des solutions de base entières.*

Démonstration. D'après le théorème 2.2, il suffit de vérifier que tous les mineurs de la matrice $A \in \Gamma$ sont égaux à 0 ou à ± 1. Soient A une matrice quelconque de Γ et Δ l'un quelconque de ses mineurs. Développons successivement le mineur Δ par rapport aux colonnes qui contiennent un seul élément non nul (1 ou —1). Il en résulte que Δ est égal à ± 1 ou que l'on obtient un mineur $\Delta' = \pm\Delta$ qui ne compte pas de colonnes à élément non nul unique. Si Δ' contient une colonne nulle,

$$\Delta = \pm\Delta' = 0.$$

S'il n'en est pas ainsi, toute colonne de Δ' compte exactement deux éléments non nuls. Soient E'_A et E''_A les ensembles de lignes de la matrice A dont l'existence est postulée par la définition de Γ. Choisissons une ligne quelconque du mineur Δ'; pour fixer les idées, considérons qu'elle appartient à E'_A. Ajoutons à la ligne choisie les autres lignes de Δ' qui entrent dans E'_A et retranchons-en toutes les lignes de Δ' qui appartiennent à E''_A.

Soit j la position arbitraire de la ligne résultante du mineur $\Delta'' = \Delta'$ obtenu. Par condition, la j-ième colonne de Δ' contient deux éléments non nuls. Si ces éléments sont identiques, ils figurent dans des lignes qui appartiennent à des ensembles différents et par suite, s'ils se trouvent dans la j-ième case de la ligne résultante avec des signes opposés, leur somme est 0. Mais si l'un d'eux vaut 1 et l'autre —1, ils figurent alors dans les lignes d'un même ensemble et tombent dans la j-ième case de la ligne résultante ayant gardé des signes différents. Par conséquent dans ce cas-là encore la j-ième case de la ligne résultante contient un zéro. Ces raisonnements montrent que Δ'' a une ligne nulle, c'est-à-dire

$$\Delta = \pm\Delta' = \pm\Delta'' = 0.$$

Ainsi, chaque mineur de la matrice A est égal à 0 ou à ± 1. Le théorème est ainsi démontré.

Il est curieux de noter que les conditions 2 et 3 énoncées lors de la définition de l'ensemble Γ caractérisent parfaitement l'ensemble des matrices A dont chaque colonne compte pas plus de deux éléments non nuls et qui engendrent les problèmes de programmation

linéaire L (C, A, B) ayant des solutions de base entières pour des C quelconques et des B entiers. On a précisément la proposition suivante, dont nous omettons la démonstration.

Théorème 2.5. *Si chaque colonne de la matrice A compte pas plus de deux éléments non nuls et si tout problème possible L (C, A, B) a des solutions en nombres entiers, la matrice A est contenue dans l'ensemble Γ.*

§ 3. Algorithme de programmation en nombres entiers

3.1. Ainsi qu'il ressort du paragraphe précédent, la classe des problèmes de programmation linéaire aux solutions en nombres entiers est extrêmement pauvre. Dans la majorité des cas, la résolution d'un problème fournit un programme optimal non entier. C'est pourquoi la discussion des problèmes de programmation en nombres entiers nécessite des méthodes spéciales. L'algorithme décrit ci-dessous a été proposé par R. Gomory ([27], [29]). Un algorithme analogue, dû également à R. Gomory [28] et conçu pour les problèmes de programmation linéaire partiellement en nombres entiers, est considéré au paragraphe suivant.

A la base des méthodes de programmation linéaire en nombres entiers on trouve l'idée suivante qui appartient, semble-t-il, à Dantzig. Supposons qu'il faille résoudre un problème de programmation linéaire dont toutes ou certaines variables sont astreintes à être entières. Cherchons le programme de base optimal du problème linéaire sans tenir compte pour le moment des conditions d'intégralité. Il se peut également que ce programme vérifie les conditions d'intégralité et constitue par conséquent la solution recherchée. Mais le plus vraisemblable est que le programme obtenu n'obéira pas à ces conditions. On se propose alors de former une contrainte linéaire qui soit vérifiée par tout programme entier (ou partiellement entier) du problème, mais ne le soit manifestement pas par la solution de base trouvée du problème linéaire. Du point de vue géométrique, ceci correspond à la construction d'un hyperplan délimitant strictement le sommet fixe de l'ensemble convexe M et les points de l'ensemble M dont les composantes respectives sont des entiers. Le système de contraintes linéaires du problème est complété par la nouvelle condition, après quoi l'on cherche la solution du problème linéaire ainsi obtenu. Si celle-ci vérifie les restrictions d'intégralité, elle constitue la solution recherchée du problème en nombres entiers, puisque l'ensemble convexe des conditions du problème linéaire contient le domaine de définition du problème à valeurs entières. Dans le cas contraire, on construit une nouvelle contrainte linéaire qui retranche la solution obtenue sans toucher

aux programmes entiers, et l'on reprend toute la procédure. Si la règle de formation des contraintes additionnelles est mise au point de façon satisfaisante, on est fondé à espérer qu'après quelques grandes itérations (suites d'opérations nécessaires à la résolution du problème linéaire) l'on trouve la solution du problème en nombres entiers, ou bien le système successif de conditions est contradictoire, c'est-à-dire que les conditions du problème initial sont incompatibles. La mise en œuvre des idées de Dantzig impose, par conséquent, la composition d'une règle de formation des contraintes additionnelles. Il convient de remarquer que parmi les règles qui semblent valables à première vue, toutes, en réalité, ne conviennent pas, c'est-à-dire ne conduisent pas à la solution du problème en nombres entiers (cf. le paragraphe 5.3).

R. Gomory est le premier qui ait réussi à réaliser de façon satisfaisante l'idée de Dantzig.

3.2. Les raisonnements qui suivent seront tous effectués pour le problème de programmation en nombres entiers qui consiste à maximiser la forme linéaire

$$L(X) = \sum_{j=1}^{n} c_j x_j \tag{3.1}$$

sous les conditions

$$\sum_{j=1}^{n} a_{ij} x_j = b_i, \quad i = 1, 2, \ldots, m; \tag{3.2}$$

$$x_j \geqslant 0, \quad j = 1, 2, \ldots, n, \tag{3.3}$$

$$x_j \text{ entier pour tout } j. \tag{3.4}$$

Les paramètres a_{ij}, b_i et c_j sont supposés entiers.

Considérons d'abord quelques relations bien simples qui rendent plus clair le sens des tableaux utilisés dans les algorithmes de programmation linéaire.

Choisissons un système arbitraire de m vecteurs linéairement indépendants des conditions du problème (3.1)-(3.3). Supposons, pour fixer les idées, que ce soient les m premiers vecteurs $A_1, A_2, \ldots, A_m$. Multiplions les deux membres de l'équation vectorielle

$$\sum_{j=1}^{n} A_j x_j = B, \tag{3.5}$$

équivalente au système (3.2), par la matrice

$$A^{-1} = (A_1, A_2, \ldots, A_m)^{-1}.$$

Si l'on pose

$$X_0 = A^{-1}B; \quad X_j = A^{-1}A_j, \quad j = 1, 2, \ldots, n,$$

(3.5) devient une équation vectorielle

$$\sum_{j=1}^{n} X_j x_j = X_0, \tag{3.6}$$

qui est à son tour équivalente au système

$$x_i + \sum_{j=m+1}^{n} x_{ij} x_j = x_{i0},\ i = 1, 2, \ldots, m,$$

ou

$$x_i = x_{i0} - \sum_{j=m+1}^{n} x_{ij} x_j,\ i = 1, 2, \ldots, m. \tag{3.7}$$

x_{ij} est ici le coefficient de décomposition du vecteur A_j ($B = A_C$) par rapport au système A, qui correspond au vecteur A_i. En portant dans la forme linéaire (3.1) les expressions des x_i résultant de (3.7), on obtient

$$L(X) = \sum_{i=1}^{m} \left(x_{i0} - \sum_{j=m+1}^{n} x_{ij} x_j\right) c_i + \sum_{j=m+1}^{n} c_j x_j =$$

$$= \sum_{i=1}^{m} c_i x_{i0} - \sum_{j=m+1}^{n} \left(\sum_{i=1}^{m} c_i x_{ij} - c_j\right) x_j = \Delta_0 - \sum_{j=m+1}^{n} \Delta_j x_j. \tag{3.8}$$

Δ_0 est ici la valeur de la forme linéaire (3.1) sur le vecteur

$$(x_{10}, x_{20}, \ldots, x_{m0}, 0, 0, \ldots, 0);$$

Δ_j, le critère du vecteur A_j par rapport au système A.

Les paramètres x_{ij} et Δ_j forment le contenu des tableaux du premier algorithme des méthodes primale et duale du simplexe. Pour la méthode primale, A est la base du programme de base courant; pour la méthode duale, A est la base du pseudoprogramme courant.

Les paramètres du tableau déterminent ainsi les expressions des variables de base et de la forme linéaire par l'intermédiaire des variables secondaires du programme ou du pseudoprogramme considérés. Chacune de ces expressions [cf. (3.7) ou (3.8)] est engendrée par la ligne respective du tableau.

3.3. Conformément à la structure des tableaux des algorithmes de programmation linéaire adoptée dans [87], les paramètres Δ_j ($j = 1, 2, \ldots, n$) figurent dans la dernière ($(m+1)$-ième) ligne de chaque tableau, ce qui impose dans [87] l'introduction de la notation évidente $\Delta_j = x_{m+1,j}$. Les algorithmes de programmation en nombres entiers sont liés à la construction successive de plusieurs nouvelles contraintes, c'est-à-dire nécessitent de modifier successivement le nombre de lignes des tableaux. C'est pourquoi, dans le cas considéré, il est plus commode de porter les paramètres

Δ_j sur la ligne supérieure des tableaux. Nous affecterons à cette ligne l'indice 0. La modification indiquée dans la structure des tableaux entraîne la notation

$$x_{0j} = \Delta_j, \quad j = 0, 1, 2, \ldots, n,$$

à laquelle nous nous tiendrons dans ce chapitre.

Nous résoudrons le problème linéaire (3.1)-(3.3) par la méthode primale ou duale du simplexe en faisant appel au premier algorithme. Supposons que la base optimale soit constituée par un système de vecteurs $A_{s_1}, A_{s_2}, \ldots, A_{s_m}$. Désignons les paramètres du dernier tableau par x_{ij}, de plus $\Delta_j = x_{0j}$.

Si toutes les composantes de base x_{i0} du programme optimal $X^{(1)}$ sont des entiers, $X^{(1)}$ est solution du problème (3.1)-(3.4).

Supposons que certains des x_{i0} soient fractionnaires. Choisissons l'un d'entre eux, x_{l0} par exemple, et formons la contrainte additionnelle en partant de la l-ième ligne du tableau optimal.

Introduisons les notations suivantes: $[x_{lj}]$ est la partie entière de x_{lj}, c'est-à-dire le plus grand entier ne dépassant pas x_{lj}; $\gamma_{lj} = x_{lj} - [x_{lj}]$, la partie fractionnaire du nombre x_{lj}.

Soit X un programme arbitraire du problème (3.1)-(3.3). Ses composantes x_j doivent être associées par l'équation

$$x_{s_l} = x_{l0} - \sum_{j \in J} x_{lj} x_j, \tag{3.9}$$

où J sont les numéros des vecteurs qui n'entrent pas dans la base du programme $X^{(1)}$.

Remplaçons dans (3.9) x_{lj} par l'expression de leurs parties entières et fractionnaires,

$$x_{lj} = [x_{lj}] + \gamma_{lj}.$$

Il vient

$$x_{s_l} - [x_{l0}] + \sum_{j \in J} [x_{lj}]\, x_j = \gamma_{l0} - \sum_{j \in J} \gamma_{lj} x_j. \tag{3.10}$$

La relation (3.10) a lieu pour n'importe quel programme X du problème (3.1)-(3.3). Supposons maintenant que X soit un programme entier de ce problème. Compte tenu de l'intégralité du premier membre de (3.10),

$$\gamma_{l0} - \sum_{j \in J} \gamma_{lj} x_j \text{ entier}$$

ou

$$\sum_{j \in J} \gamma_{lj} x_j = \gamma_{l0} - \zeta, \ \zeta \text{ entier}. \tag{3.11}$$

D'après la définition de la partie fractionnaire,

$$0 \leqslant \gamma_{lj} < 1.$$

Par conséquent,

$$\sum_{j\in J} \gamma_{lj}x_j \geqslant 0, \quad \gamma_{l0} < 1. \tag{3.12}$$

Si l'on suppose que le nombre entier ζ de la ormule (3.11) est négatif ($\zeta \leqslant -1$), on aboutit à l'inégalité

$$\sum_{j\in J} \gamma_{lj}x_j + 1 \leqslant \gamma_{l0},$$

qui avec la première inégalité (3.12) conduit à la restriction

$$1 \leqslant \gamma_{l0},$$

qui contredit la deuxième condition (3.12).

Il faut donc admettre que le nombre ζ est non négatif, c'est-à-dire reconnaître que l'inégalité

$$\sum_{j\in J} \gamma_{lj}x_j \geqslant \gamma_{l0} \tag{3.13}$$

se vérifie quel que soit le programme en nombres entiers du problème (3.1)-(3.3). L'inégalité obtenue est la contrainte additionnelle recherchée. Il est aisé de s'assurer que le programme $X^{(1)}$, déterminé au terme de la résolution du problème (3.1)-(3.3), ne satisfait pas à (3.13), c'est-à-dire que la contrainte additionnelle ainsi construite retranche effectivement une certaine partie de l'ensemble convexe initial.

En effet, pour le programme $X^{(1)}$, le premier membre de (3.13) est nul et le second membre de cette inégalité est un nombre positif en tant que partie fractionnaire de la grandeur non entière x_{l0}.

Notons qu'il est possible de prendre également 0 pour l, c'est-à-dire, si $x_{0,0}$ est un nombre fractionnaire, former la contrainte additionnelle en partant de la forme linéaire du problème transformée

$$L(X) = x_0 = x_{0.0} - \sum_{j\in J} x_{0,j}x_j.$$

En effet, l'intégralité des coefficients c_j assure celle de la variable $x_0 = L(X)$ pour tout programme entier X. Par conséquent, tous les raisonnements faits pour $1 \leqslant \tilde{l} \leqslant m$ restent valables également pour $l = 0$.

La contrainte (3.13), qu'on peut écrire sous la forme équivalente

$$x_{n+1} = -\gamma_{l0} + \sum_{j\in J} \gamma_{lj}x_j, \quad x_{n+1} \geqslant 0, \tag{3.14}$$

est ajoutée au système de contraintes linéaires (3.2)-(3.3), après quoi on procède à la résolution du problème (3.1)-(3.3), (3.14). Les variables du nouveau problème sont $x_1, x_2, \ldots, x_n$ et x_{n+1}. Les conditions de ce problème sont déjà établies par rapport à

$x_{s_1}, x_{s_2}, \ldots, x_{s_m}$ (variables de base du programme $X^{(1)}$) et à la nouvelle variable x_{n+1}, alors que la forme linéaire est exprimée à l'aide des variables secondaires du programme $X^{(1)}$. Tenant compte de ce que les paramètres x_{0j} sont non négatifs, on en conclut que la représentation disponible correspond au pseudoprogramme du problème (3.1)-(3.3), (3.14) aux composantes de base

$$x_{s_i} = x_{i0},\ i = 1, 2, \ldots, m;\ x_{n+1} = -\gamma_{l0}.$$

Le tableau de ce pseudoprogramme est formé en bordant le tableau correspondant à $X^{(1)}$ d'une ligne composée d'éléments

$$x_{m+1,j} = \begin{cases} -\gamma_{l0}, & j = 0, \\ -\gamma_{lj}, & j \in J, \\ 1, & j = n+1, \\ 0 \text{ dans les autres cas} \end{cases}$$

et d'une colonne unité e_{n+1}.

Pour résoudre le problème (3.1)-(3.3), (3.14), il est logique de faire appel à la méthode duale du simplexe. La première itération s'achève nécessairement par l'élimination de la base du vecteur qui correspond à la nouvelle variable x_{n+1}, puisque la première colonne du tableau ne compte qu'un seul élément négatif, $x_{m+1,0} = -\gamma_{l0}$. Après quelques itérations de la méthode duale du simplexe, on obtient la solution du problème (3.1)-(3.3), (3.14) ou bien l'on établit qu'il est irréalisable à cause de l'incompatibilité de ses conditions. Cette dernière éventualité indique que le problème en nombres entiers initial n'a pas de solution, puisque, comme on l'a établi, chaque programme entier vérifie nécessairement le système (3.2), (3.3), (3.14).

Supposons que l'analyse du problème (3.1)-(3.3), (3.14) s'achève par la construction d'un tableau optimal correspondant à la solution $X^{(2)}$. Si les éléments de la première colonne du tableau ainsi obtenu sont des entiers, les n premières composantes de $X^{(2)}$ forment alors la solution recherchée puisque chaque programme du problème (3.1)-(3.4), étant complété par la valeur correspondante de la variable x_{n+1}, constitue un programme du problème (3.1)-(3.3), (3.14) avec la même valeur de la forme linéaire. Mais si parmi les composantes de base de la solution $X^{(2)}$ il y a des nombres fractionnaires, alors, conformément à la règle déjà décrite, l'une des lignes du tableau optimal engendre une nouvelle contrainte additionnelle, le tableau optimal est bordé d'une ligne et d'une colonne unité, et l'on applique de nouveau la méthode duale du simplexe. Nous conviendrons d'appeler *grande* itération la succession des transformations nécessaires soit pour obtenir à partir du tableau optimal disponible, à éléments fractionnaires dans la première colonne, le tableau optimal suivant, soit pour mettre en évidence l'incompatibilité des contraintes correspondantes. Chaque grande itération se compose d'opérations préalables, dont le but est d'étendre d'une

ligne et d'une colonne le tableau optimal disponible, et d'une série d'itérations de la méthode duale du simplexe.

Suivant le résultat de la grande itération, nous distinguerons les cas suivants.

1°. Tout élément de la première colonne du tableau optimal est un entier.

2°. La méthode duale conduit à un tableau intermédiaire dont tous les éléments d'une des lignes sont non négatifs, à l'exception du premier élément négatif (il s'agit des lignes à indice $i \geqslant 1$).

3°. La première colonne du tableau optimal compte des nombres fractionnaires.

Le cas 1° fixe l'obtention du programme optimal du problème (3.1)-(3.4).

Le cas 2° traduit l'incompatibilité des conditions (3.2)-(3.4). Enfin, le cas 3° impose le passage à la grande itération suivante.

Appelons par convention *additionnelles* les variables x_{n+t} introduites dans le problème au début de chaque grande itération, et *principales* les variables x_j avec $1 \leqslant j \leqslant n$.

Si la variable additionnelle x_{n+t} est une variable secondaire pour un certain pseudoprogramme intermédiaire, l'équation $x_{n+t} = 0$ fait partie du nombre de relations qui définissent ce pseudoprogramme. Dès que la variable x_{n+t} devient variable de base, sa valeur perd son importance pour les variables principales du problème; la ligne et la colonne du tableau courant, qui correspondent à x_{n+t}, sont alors rayées. Voici l'interprétation géométrique de cette règle de réduction du tableau.

Dès que le pseudoprogramme courant se trouve à l'intérieur du demi-espace $x_{n+t} \geqslant 0$, la contrainte additionnelle définie par l'hyperplan $x_{n+t} = 0$ est supprimée. La règle de réduction des tableaux délimite leurs dimensions à n lignes et $2n - m$ colonnes. En effet, le nombre maximal possible de composantes de base est défini par le nombre n des variables principales. C'est le cas extrême auquel correspondent $n - m$ variables additionnelles.

Ainsi, la méthode considérée de programmation en nombres entiers s'apparente, quant à son algorithme, à la méthode duale du simplexe et diffère de cette dernière par les règles d'extension et de réduction des tableaux courants.

Dans des conditions déterminées on peut garantir le caractère fini du processus décrit, c'est-à-dire la possibilité d'obtenir la solution du problème en nombres entiers ou de mettre en evidence son irrésolubilité, le tout en un nombre fini d'itérations.

Les justifications correspondantes font l'objet de deux paragraphes suivants.

3.4. Numérotons de façon arbitraire les variables principales du problème et faisons correspondre à chaque vecteur

$$X = (x_1, x_2, \ldots, x_n, x_{n+i_1}, x_{n+i_2}, \ldots, x_{n+i_r})$$

le vecteur

$$\vec{L}(X) = (x_0, x_1, x_2, \ldots, x_n),$$

où

$$x_0 = L(X) = \sum_{j=1}^{n} c_j x_j.$$

Nous conviendrons de considérer que $\vec{L}(X_1)$ est supérieur à $\vec{L}(X_2)$ et d'écrire $\vec{L}(X_1) > \vec{L}(X_2)$ si

$$x_0^{(1)} > x_0^{(2)},$$

ou avec un certain $0 < \mu \leqslant n - 1$

$$x_j^{(1)} = x_j^{(2)} \quad \text{pour} \quad j \leqslant \mu,$$
$$x_{\mu+1}^{(1)} > x_{\mu+1}^{(2)}.$$

Si $x_j^{(1)} = x_j^{(2)}$ pour tout $j = 0, 1, 2, \ldots, n$, $\vec{L}(X_1)$ est considéré naturellement égal à $\vec{L}(X_2)$.

Cette méthode de comparaison conduit à la mise en ordre du système des vecteurs $\vec{L}(X)$. En d'autres termes, n'importe quels $\vec{L}(X_1) \neq \vec{L}(X_2)$ donnent lieu à l'une des deux alternatives

$$\vec{L}(X_1) < \vec{L}(X_2) \quad \text{ou} \quad \vec{L}(X_1) > \vec{L}(X_2).$$

De plus, les conditions $\vec{L}(X_1) < \vec{L}(X_2) < \vec{L}(X_3)$ entraînent la relation $\vec{L}(X_1) < \vec{L}(X_3)$. Ce procédé de mise en ordre s'appelle parfois lexicographique, et l'écriture $\vec{L}(X_1) < \vec{L}(X_2)$ se lit: le vecteur $\vec{L}(X_1)$ est lexicographiquement inférieur au vecteur $\vec{L}(X_2)$. Dans ce qui suit, nous dirons que l'un des vecteurs est « supérieur » ou « inférieur » en omettant le mot « lexicographiquement ».

En se servant de la notion introduite qui permet de choisir le plus grand entre deux vecteurs $\vec{L}(X_1)$ et $\vec{L}(X_2)$, il est possible d'étudier le problème de maximisation de la forme linéaire vectorielle $\vec{L}(X)$ sous les conditions linéaires par rapport aux composantes de X. Pour résoudre un tel problème on peut faire appel aux méthodes usuelles de programmation linéaire, à cette différence près que les critères des vecteurs des conditions sont dans le cas présent des grandeurs vectorielles, et leur comparaison impose l'utilisation de la règle énoncée ci-dessus. Remplaçons dans le problème (3.1)-(3.4) la forme linéaire (3.1) par la forme linéaire vectorielle

$$\vec{L}(X) = (L(X), x_1, x_2, \ldots, x_n). \tag{3.15}$$

Le sens de cet artifice est qu'il se trouve que le problème de maximisation de $\vec{L}(X)$ sous les conditions (3.2)-(3.3) et pour un

nombre arbitraire de contraintes supplémentaires du type (3.14) ne peut admettre plus d'une solution. C'est pourquoi, en retranchant au début de chaque grande itération la solution du problème précédent, nous pouvons avoir la certitude de la décroissance monotone de (3.15), et par conséquent du fait que la méthode est finie.

Introduisons les formules des critères vectoriels $\vec{\Delta}_j = \vec{x}_{0j}$. Considérons le problème correspondant à une grande itération arbitraire et un système Γ complet quelconque de vecteurs des conditions linéairement indépendants $R_1, R_2, \ldots, R_\alpha, R_{n+1}, R_{n+2}, \ldots$ $\ldots, R_{n+\beta}$ $(\alpha + \beta \geqslant m,\ \alpha \leqslant n)$ de ce problème. Les vecteurs R_i correspondent aux variables principales pour $1 \leqslant i \leqslant n$ et aux variables additionnelles x_{s_i} pour $i \geqslant n + 1$.

La première composante $x_{0j}^{(0)}$ du vecteur $\vec{x}_{0j}$ est évidemment égale à x_{0j}. Pour des coefficients donnés d_j de la forme linéaire, nuls pour toutes les variables additionnelles, le critère du vecteur des conditions R_j aux composantes x_{ij} $(i = 1, 2, \ldots, \alpha + \beta)$ de décomposition par rapport à Γ peut être calculé, comme on le sait, d'après la formule

$$\Delta_j^{(d)} = \sum_{i=1}^{\alpha} d_{s_i} x_{ij} - d_j. \tag{3.16}$$

Pour déterminer la composante $x_{0j}^{(\lambda)}$ du vecteur $\vec{x}_{0j}$, il faut remplacer dans (3.16) le vecteur d par le vecteur unité e_λ $(\lambda = 1, 2, \ldots, n)$.

Finalement, si $R_j \notin \Gamma$, on a

$$x_{0j}^{(\lambda)} = \begin{cases} x_{ij}, & \text{si} \quad \lambda = s_i, \quad i = 1, 2, \ldots, \alpha, \\ -1, & \text{si} \quad \lambda = j, \\ 0 & \text{dans les autres cas.} \end{cases} \tag{3.17}$$

On voit que les critères des vecteurs $\vec{x}_{0j}$ sont parfaitement définis par les paramètres x_{0j} et x_{ij} du tableau respectif. Vérifions que pour tout vecteur des conditions $R_j \notin \Gamma$ le critère $\vec{x}_{0j}$ diffère de zéro. En effet, avec $j \leqslant n$, cette conclusion résulte immédiatement de (3.17) du fait que la j-ième composante du vecteur des critères est égale à -1. Supposons maintenant que $j \geqslant n + 1$, c'est-à-dire que R_j corresponde à une certaine variable additionnelle. Le type des conditions supplémentaires entraîne immédiatement que les valeurs des variables additionnelles sont déterminées univoquement par celles des variables principales. Par suite, tout système constitué de vecteurs R_k, avec $k \geqslant n + 1$, est linéairement indépendant, et donc, parmi les nombres x_{ij} $(i = 1, 2, \ldots, \alpha)$, il en est nécessairement qui sont non nuls.

Nous avons démontré ainsi la proposition suivante.

Lemme 3.1. *Le problème à forme linéaire* (3.15) *correspondant à une grande itération quelconque est non dégénéré au sens dual* (*c'est-à-dire son dual est non dégénéré*).

La résolution du problème en nombres entiers étant recherchée à l'aide de la méthode duale du simplexe, chaque itération, d'après le lemme 3.1, conduit à la décroissance de $\vec{L}(X)$.

Notons que le remplacement de $L(X)$ par $\vec{L}(X)$ ne se fait pratiquement sentir que par l'utilisation, lors de la définition du vecteur à ajouter à la base, non seulement de la ligne Δ, mais aussi d'autres lignes du tableau, en se guidant par la formule (3.17). Ceci concerne en particulier la grande itération initiale qui amène la solution du problème (3.1)-(3.3). Supposons dans ce qui suit que le choix du vecteur à inclure dans la base s'effectue à chaque étape en partant des critères vectoriels, ce qui conduit à une décroissance monotone de la fonction $\vec{L}(X)$. Pour rendre la solution du problème (3.15), (3.2), (3.3) réalisable, exigeons que le problème (3.1)-(3.3) soit possible et que l'ensemble de ses programmes optimaux soit borné.

3.5. Nous sommes à même maintenant de démontrer que la méthode est finie. Raisonnons par l'absurde et supposons que le cas 3° peut revenir dans un nombre infini de grandes itérations.

Si l'on admet que la forme linéaire (3.1) est bornée inférieurement sur l'ensemble (3.2), (3.3), chaque composante de la suite des valeurs $\vec{L}(X^{(\mu)})$ (à décroissance monotone) de la fonction vectorielle $\vec{L}(X)$ est bornée inférieurement. On entend ici par $X^{(\mu)}$ le résultat de la μ-ème grande itération.

Soit r le numéro minimal tel que la suite $\{x_r^{(\mu)}\}$, constituée des r-ièmes composantes de $\{\vec{L}(X^{(\mu)})\}$, compte un nombre infini d'éléments différents.

Il vient

$$\vec{L}(X^{(\mu)}) > \vec{L}(X^{(\mu+1)}), \quad \mu = 1, 2, \ldots, \tag{3.18}$$

$$x_i^{(\mu)} = x_i^* \quad \text{pour} \quad i < r, \quad \mu \geqslant \mu_0, \tag{3.19}$$

où μ_0 est un nombre suffisamment grand.

(3.18) et (3.19) donnent

$$x_r^{(\mu)} \geqslant x_r^{(\mu+1)} > x_r^* \quad \text{avec} \quad \mu \geqslant \mu_0, \tag{3.20}$$

où $x_r^* = \lim\limits_{\mu\to\infty} x_r^{(\mu)}$.

Convenons de construire la contrainte supplémentaire en partant de celle des lignes du tableau au premier élément fractionnaire qui correspond à la composante de base affectée d'un numéro mini-

mal (la ligne supérieure du tableau correspond toujours à la composante x_0, c'est-à-dire est révisée en premier lieu).

Supposons que la contrainte supplémentaire formée suivant la l-ième ligne du tableau s'écrit

$$x_{n+\mu} - \sum_{j \in J} \gamma_{lj} x_j = -\gamma_{l0}, \quad x_{n+\mu} \geqslant 0. \tag{3.21}$$

la première étape de la grande itération successive s'achève par l'élimination du nombre des variables de base d'une nouvelle variable $x_{n+\mu}$, remplacée par une certaine variable x_λ. La valeur de la variable x_{s_l} associée à la l-ième ligne du tableau devient

$$x'_{0l} = x_{0l} - \frac{\gamma_{l0} \cdot x_{l\lambda}}{\gamma_{l\lambda}}, \tag{3.22}$$

où x_{0l} est son ancienne valeur, $\gamma_{l\lambda} > 0$ la partie fractionnaire de $x_{l\lambda}$.

En utilisant (3.22), on vérifie aisément qu'avec $i < r$, x_i^* est un entier. En effet, s'il n'en est pas ainsi et si $i_0 < r$ est l'indice minimal i tel que x_i^* soit un nombre fractionnaire, alors à la μ-ème grande itération ($\mu > \mu_0$) la contrainte supplémentaire est engendrée par la ligne l correspondant à la variable x_{i_0}, et, d'après la formule (3.22), la valeur de cette variable, après la première étape de l'itération, cesse d'être égale à $x_{i_0}^*$.

Tenant compte ensuite de la décroissance monotone de la fonction vectorielle à minimiser $\vec{L}(X)$, on obtient pour un certain $i' \leqslant i_0 < r$

$$x_{i'}^{(\mu+1)} < x_{i'}^{(\mu)} = x_{i'}^*,$$

ce qui contredit (3.19).

La convergence de la suite $\{x_r^{(\mu)}\}$ vers x_r^* et les inégalités (3.20) entraînent la relation

$$[x_r^{(\mu)}] = [x_r^*] < x_r^{(\mu)} < [x_r^{(\mu)}] + 1, \tag{3.23}$$

qui est vraie pour tout $\mu \geqslant \mu'$, où μ' est un nombre suffisamment grand.

Ainsi, pour tout $\mu \geqslant \max(\mu_0, \mu')$, le nombre $x_r^{(\mu)}$ est une composante fractionnaire du vecteur $X^{(\mu)}$ à numéro minimal. Considérons le début de la $(\mu + 1)$-ème grande itération, μ étant suffisamment grand. D'après la règle mentionnée plus haut, la contrainte supplémentaire de cette itération est déterminée par la ligne l correspondant à la composante x_r.

Supposons qu'à la première étape de l'itération, la variable x_λ entre dans la base en prenant la place de la variable additionnelle $x_{n+\mu+1}$ qui vient d'être introduite. D'après la formule (3.22) et

la relation (3.23), la valeur de la variable x_r devient

$$x_r^{(\mu,\,1)} = x_r^{(\mu)} - (x_r^{(\mu)} - [x_r^*]) \cdot \frac{x_{l\lambda}}{\gamma_{l\lambda}}. \tag{3.24}$$

La positivité de $\gamma_{l\lambda}$ se déduit de la règle de choix de l'indice λ. Donc, $x_{l\lambda} \neq 0$ en tant que grandeur admettant une partie fractionnaire positive.

Si $x_{l\lambda}$ était un nombre négatif, (3.24) entraînerait l'inégalité

$$x_r^{(\mu,\,1)} > x_r^{(\mu)}$$

qui, en vertu de la décroissance monotone de $\vec{L}(X)$, conduit à la relation

$$x_i^{(\mu+1)} < x_i^{(\mu)} \quad \text{pour un certain} \quad i < r,$$

incompatible avec la condition (3.19). Par conséquent,

$$x_{l\lambda} > 0 \quad \text{et} \quad \gamma_{l\lambda} = x_{l\lambda} - [x_{l\lambda}] \leqslant x_{l\lambda}.$$

Cette dernière relation permet d'obtenir à partir de (3.24) l'inégalité

$$x_r^{(\mu,\,1)} \leqslant x_r^{(\mu)} - (x_r^{(\mu)} - [x_r^*]) = [x_r^*],$$

qui en vertu de (3.19) et de la décroissance monotone de la fonction $\vec{L}(X)$, donne l'inégalité

$$x_r^{(\mu+1)} \leqslant x_r^{(\mu,\,1)} \leqslant [x_r^*],$$

incompatible avec la condition (3.23).

Ainsi, l'hypothèse de la possibilité d'un nombre infini de grandes itérations nous a menés à une contradiction. Le caractère fini de la méthode est démontré. La conclusion obtenue peut être énoncée de la façon suivante.

Théorème 3.1. *Si la forme linéaire* (3.1) *est bornée supérieurement et inférieurement sur l'ensemble convexe* (3.2)-(3.3), *l'ensemble des programmes optimaux du problème* (3.1)-(3.3) *étant un polyèdre convexe, le choix du vecteur à ajouter à la base se faisant à l'aide des critères vectoriels* $\vec{x}_{0j}$ *et la contrainte supplémentaire se formant à partir de la ligne du tableau correspondant à la composante de base fractionnaire de numéro minimal, alors l'algorithme de programmation en nombres entiers est fini.*

3.6. Considérons celles des hypothèses du théorème 3.1 dont l'observation assure la finitude de la méthode. La première hypothèse (forme linéaire bornée du problème sur l'ensemble convexe initial et l'ensemble borné des programmes optimaux du problème (3.1)-(3.3)) est vérifiée dans la grande majorité des problèmes pratiques.

L'hypothèse que le problème (3.1)-(3.3) est résoluble et que

l'ensemble de ses programmes optimaux est borné est indispensable pour faire aboutir l'exploration du problème (3.15), (3.2), (3.3) à la construction de son programme optimal ou à la mise en évidence de l'incompatibilité de ses conditions. Remarquons que l'hypothèse de l'ensemble borné des solutions du problème (3.1)-(3.3) peut être levée au prix de l'introduction d'une ligne supplémentaire dans les tableaux de l'algorithme. L'hypothèse de la forme linéaire (3.1) bornée inférieurement ne s'emploie que pour garantir l'allure bornée de la suite $\{x_0^{(\mu)}\}$. Constatons que la relation $\lim\limits_{\mu\to\infty} x_0^{(\mu)} = -\infty$ n'est possible que lorsque les conditions (3.2)-(3.4) sont incompatibles. Par conséquent, lorsque ces conditions sont compatibles, l'hypothèse du théorème 3.1 suivant laquelle la forme linéaire (3.1) est bornée inférieurement sur l'ensemble (3.2)-(3.3) peut être levée sans préjudice pour le caractère fini de l'algorithme.

La deuxième hypothèse du théorème 3.1, qui exige de choisir le vecteur destiné à être inclus dans la base en se guidant par le critère $\vec{x}_{0j}$, a pour objet d'éviter le cyclage. Etant donné qu'en programmation linéaire, le cyclage est un phénomène rare, on peut, dans les calculs pratiques, se servir des critères usuels $x_{0j} = \Delta_j$, en rendant univoque le choix du vecteur à ajouter par l'un des procédés les plus simples (cf. [87], paragraphe 3.2 du chapitre 9).

Toutefois, compte tenu d'expériences numériques peu nombreuses dans le domaine de l'utilisation de l'algorithme, il convient de ne considérer cette dernière recommandation que comme une recommandation préliminaire. Il se peut que l'expérience accumulée dans le calcul rende nécessaire, dans certaines étapes de la procédure, de choisir le vecteur à introduire à l'aide d'une règle susceptible de rendre le cyclage tout à fait impossible.

La troisième hypothèse du théorème 3.1, liée à la règle de choix de la ligne du tableau, qui engendre la contrainte additionnelle peut être affaiblie de la façon suivante.

Choisissons un entier naturel N suffisamment grand. L'ordre de formation de la contrainte additionnelle peut être quelconque. Toutefois, s'il s'avère que telle variable (y compris x_0) prend au cours de N itérations successives des valeurs fractionnaires, et que la ligne correspondante ne participe pas à la construction des contraintes additionnelles, alors la nouvelle contrainte additionnelle doit être formée à partir de la ligne associée à cette variable. Le passage à la règle indiquée plus souple conserve le caractère fini de la méthode. (La conduite des raisonnements respectifs fait l'objet de l'exercice 7.) Le choix de la ligne nécessaire pour former la contrainte additionnelle peut se faire de sorte que la première étape de la grande itération aboutisse à une décroissance maximale de la forme linéaire. Ceci signifie que l'indice l doit être choisi

à partir de la relation

$$\frac{x_{0\lambda(l)}}{\gamma_{l\lambda(l)}}\gamma_{l0}=\max_i\left[\frac{x_{0\lambda(i)}}{\gamma_{i\lambda(i)}}\gamma_{i0}\right], \tag{3.25}$$

où $\lambda(i)$ est défini par la condition

$$\frac{x_{0\lambda(i)}}{\gamma_{i\lambda(i)}}=\min_{\gamma_{ij}>0}\frac{x_{0j}}{\gamma_{ij}}, \quad \gamma_{i\lambda(i)}>0. \tag{3.26}$$

Une règle plus simple de choix de l consiste à négliger la façon dont le rapport $\frac{x_{0\lambda(i)}}{\gamma_{i\lambda(i)}}$ dépend de i:

$$\gamma_{l0}=\max_i \gamma_{i0}. \tag{3.27}$$

Si l'on néglige la façon dont $\lambda(i)$ dépend de i, on obtient une règle intermédiaire quant à sa complexité:

l est choisi d'après (3.25), mais $\lambda(i)$ est remplacé par un indice fixe λ, que l'on détermine à partir de la condition

$$x_{0\lambda}=\min_j x_{0j}. \tag{3.28}$$

Dans ce qui suit, appelons respectivement première, deuxième et troisième règle les règles ci-dessus énoncées.

Il semble que la rapidité avec laquelle la méthode converge croît en moyenne à mesure que se complique la règle de choix de l. Pourtant la modique expérience des calculs accumulée jusqu'à ces derniers temps dans l'application de l'algorithme ne permet pas de recommander avec certitude l'utilisation de l'une quelconque de ces règles. D'une façon générale, il convient de noter que l'algorithme de Gomory est encore peu étudié sous son aspect numérique. On connaît des exemples de problèmes de taille modeste dont la résolution demande un nombre énorme d'itérations. Mais d'autre part il y eut également des problèmes de grande taille dont l'exploration avait exigé des calculs peu importants. Si en programmation linéaire on observe une dépendance relativement rigoureuse entre le nombre de conditions-égalités du problème et celui des itérations nécessaires à sa résolution, cette dépendance n'existe pas pour les problèmes en nombres entiers. Le nombre de variables du problème est également peu lié au volume des calculs à faire. Il semble que la grande quantité de travail à fournir dans le cas d'un problème en nombres entiers est déterminée non seulement par sa taille, mais encore par des particularités de structure plus fines du polyèdre des conditions, propriétés liées à la distribution dans ce polyèdre des points entiers. La mise en évidence de ces particularités présente un grand intérêt théorique et pratique. Lors de la résolution d'un problème en nombres entiers, le nombre d'itérations peut être très

grand. C'est pourquoi, lors de la composition de programmes pour ordinateurs, il convient d'accorder une attention particulière aux mesures permettant de réduire le rôle négatif des erreurs d'arrondi. La mise au point d'un système de contrôle efficace est l'une des conditions décisives pour rendre le programme opérant.

3.7. Voici un exemple numérique pour illustrer l'ordre de résolution d'un problème en nombres entiers par la méthode exposée. Ce problème consiste à maximiser la forme linéaire

$$L(X) = 2x_1 + x_2 + 4x_3 + 3x_5 + x_6 + 2x_7 + 7x_8 + 3x_9 \tag{3.29}$$

sous les conditions

$$\left.\begin{aligned} x_1 + 3x_2 + 2x_3 + 4x_4 + x_5 + 5x_6 + 3x_7 + 2x_8 + 6x_9 &= 14, \\ 4x_1 + 5x_2 + 4x_3 + x_4 + 6x_5 + 2x_6 + 4x_7 + 7x_8 + 3x_9 &= 8, \end{aligned}\right\} \tag{3.30}$$

$$x_j \geqslant 0, \quad j = 1, 2, \ldots, 9, \tag{3.31}$$

$$x_j \text{ entier pour } j = 1, 2, \ldots, 9. \tag{3.32}$$

Nous n'allons pas examiner ici l'étape préalable de l'algorithme qui consiste à résoudre le problème (3.29)-(3.31). Notons seulement que la base optimale de ce problème comporte des vecteurs A_8 et A_9.

La procédure de résolution du problème en nombres entiers (3.29)-(3.32) est portée sur les tableaux 5.1 (0 à 11), dont le premier est affecté d'après l'usage du numéro 0. Chaque tableau, sauf le tableau 0, correspond à une étape de la méthode duale du simplexe. Les paramètres qui correspondent à la base optimale (A_8, A_9) figurent dans les trois premières lignes du tableau 0. Comme la colonne A_0 contient des éléments fractionnaires, il faut introduire une contrainte supplémentaire. Dans le cas concerné, celle-ci peut être formée d'après la première ou la deuxième ligne du tableau 0. Le critère $x_{03} = \Delta_3 = 0$, et chacun des nombres x_{13} et x_{23} est fractionnaire. C'est pourquoi la règle 1 du paragraphe 3.6 ne permet pas de donner la préférence à l'une de ces lignes. Construisons la contrainte supplémentaire à partir des éléments de la première ligne. A cette fin, déterminons les parties fractionnaires de ces éléments et portons-les avec le signe moins dans les cases respectives de la 3-ième ligne. Ajoutons en outre au tableau 0 la colonne unité $A_{10} = e_4$, avec l'unité à la 3-ième ligne. Le vecteur A_{10} associé à la variable additionnelle x_{10} est retranché de la pseudobase. Pour choisir le vecteur à inclure dans la pseudobase, on compose comme d'habitude la ligne θ, en y portant les rapports

$$-\frac{x_{j0}}{x_{j3}} \text{ avec } x_{j3} < 0.$$

Le minimum de la ligne θ s'obtient avec $j = 3$; le vecteur A_3 doit occuper dans la pseudobase la place du vecteur A_{10}. La ligne 3 et la colonne A_3 sont directrices pour la transformation ultérieure du tableau, dont le résultat est inscrit dans les quatre premières lignes de la colonne du tableau 1. Notons que la première étape n'a pas conduit à une modification de la forme linéaire. Les cases 1 à 3 de la colonne A_0 du tableau 1 contiennent des éléments non négatifs. Par suite, l'itération effectuée est unique dans la première grande itération, et elle a abouti au résultat 3°. La grande itération suivante débute par le choix de la ligne du tableau 1 qui engendre la nouvelle contrainte additionnelle. Nous avons de nouveau deux possibilités; nous choisissons la ligne 3. Dans la 4-ième ligne du tableau 1 sont écrites les parties fractionnaires des éléments correspondants de la ligne 3 avec le signe moins. Le tableau 1 est complété par le vecteur unité $e_8 = A_{11}$. La ligne nouvellement remplie est directrice. La ligne θ permet de déterminer la colonne directrice, qui s'avère être A_6. On

Tableau 5.1

		c_j	—	—	2	1	4		3	1	2	7	3				—
	N	C_X	B_X	A_0	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}	A_{11}	A_{12}	Nombre d'itérat.
	0	—	Δ	8	2	4		1	3	1	2						
+	1	7	A_8	1/6	7/12	7/12	1/2	—1/6	11/12	—1/12	5/12	1					
	2	3	A_9	41/18	—1/36	11/36	1/6	13/18	—5/36	31/36	13/36		1				0
←	3		A_{10}	—1/6	—7/12	—7/12	$\boxed{-1/2}$	--5/6	—11/12	—11/12	—5/12			1			
	4	—	0	—	24/7	48/7	0	6/5	36/11	12/11	24/5						
	0	—	Δ	8	2	4		1	3	1	2						
	1	7	A_8					—1		—1		1		1			
	2	3	A_9	20/9	—2/9	1/9		4/9	—4/9	5/9	2/9		1	1/3			1
+ →	3	4	A_3	1/3	7/6	7/6	1	5/3	11/6	11/6	5/6			—2			
←	4		A_{11}	—1/3	—1/6	—1/6		—2/3	—5/6	$\boxed{-5/6}$	—5/6				1		
	5	—	$0'_4$	—	12	24		3/2	18/5	6/5	12/5						

	0	—	Δ	38/5	9/5	19/5		1/5	2		1				6/5		
	1	7	A_8	2/5	1/5	1/5		—1/5	1		1	1		1	—6/5		
	2	3	A	2	—1/3				—1		—1/3		1	1/3	2/3		2
←	3	4	A_3	—2/5	4/5	4/5	1	1/5			—1			$\boxed{-2}$	11/5		
→	4	1	A_6	2/5	1/5	1/5		4/5	1	1	1				—6/5		
	5	—	θ	—							1			0			
+	0	—	Δ	38/5	9/5	19/5		1/5	2		1				6/5		
	1	7	A_8	1/5	3/5	3/5	1/2	—1/10	1		1/2	1			1/10		
	2	3	A_9	29/15	—1/5	2/15	1/6	1/30	—1		—1/2		1		31/30		
→	3		A_{10}	1/5	—2/5	—2/5	—1/2	—1/10			1/2			1	—11/10		3
	4	1	A_6	2/5	1/5	1/5		4/5	1	1	1				—6/5		
←	5		A_{12}	—3/5	—4/5	—4/5		$\boxed{-1/5}$							—1/5	1	
	6	—	θ	—	9/4	19/4		1							6		

Suite

		c_j	—	—	2	1	4		3	1	2	7	3				—
	N	C_X	B_X	A_0	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{11}	A_{12}	A_{13}	Nombre d'itérat.
	0	—	Δ	7	1	3			2		1			1	1		
	1	7	A_8	1/2	1	1	1/2		1		1/2	1			—1/2		
	2	3	A_9	11/6	—1/3		1/6		—1		—1/2		1	1	1/6		
←	3	1	A_6	—2	$\boxed{-3}$	—3			1	1	1			—2	4		4
→	°4		A_4	3	4	4		1						1	—5		
	5	—	0	—	1/3	1								1/2			
	0	—	Δ	19/3		2			7/3	1/3	4/3			1/3	7/3		
←	1	7	A_8	—1/6			1/2		4/3	1/3	5/6	1		$\boxed{-2/3}$	5/6		
	2	3	A_9	37/18		1/3	1/6		—10/9	—1/9	—11/18		1	11/9	—5/18		
→	3	2	A_1	2/3	1	1			—1/3	—1/3	—1/3			2/3	—4/3		5
	4		A_4	1/3				1	4/3	4/3	4/3			—5/3	1/3		
	5	—	0	—										1/2			

	0	—	Δ	25/4		2	1/4		3	1/2	7/4	1/2			11/4		
→	1		A_{11}	1/4			—3/4		—2	—1/2	—5/4	—3/2		1	—5/4		
+	2	3	A_9	7/4		1/3	13/12		4/3	1/2	11/12	11/6	1		5/4		
	3	2	A_1	1/2	1	1	1/2		1		1/2	1			—1/2		6
	4		A_4	3/4			—5/4	1	—2	1/2	—3/4	—5/2			—7/4		
←	5		A_{13}	—3/4		—1/3	—1/12		—1/3	—1/2	—11/12	$\boxed{-5/6}$			—1/4	1	
	6	—	0	—		6	3		9	1	21/11	3/5			11		
	0	—	Δ	29/5		9/5	1/5		14/5	1/5	6/5				13/5	3/5	
	1	3	A_9	1/10		—2/5	9/10		3/5	—3/5	—11/10		1		7/10	11/5	
←	2	2	A_1	—2/5	1	3/5	2/5		3/5	$\boxed{-3/5}$	—3/5				—4/5	6/5	
	3		A_4	3		1	—1	1	—1	2	2	1			—1	—3	7
→	4	7	A_8	9/10		2/5	1/10		2/5	3/5	11/10	1			3/10	—6/5	
	5	—	0	—						1/3	2				13/4		

Suite

		c_j	—	—	2	1	4		3	1	2	7	3				—
	N	C_X	B_X	A_0	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{12}	A_{13}	A_{14}	Nombre d'itérat.
+	0	—	Δ	17/3	1/3	2	1/3		3		1			7/3	1		
	1	3	A_9	1/2	—1	—1	1/2				—1/2		1	3/2	1		
→	2	1	A_6	2/3	—5/3	—1	—2/3		—1	1	1			4/3	—2		
	3		A_4	8/3	10/3	3	1/3	1	1					—11/3	1		8
	4	7	A_8	1/2	1	1	1/2		1		1/2	1		—1/2			
←	5		A_{14}	—2/3	—1/3		$\boxed{-1/3}$							—1/3		1	
	6	—	θ	—	1		1							7			
	0	—	Δ	5		2			3		1			2	1	1	
	1	3	A_9	—1/2	—3/2	—1					—1/2		1	1	1	3/2	
	2	1	A_6	2	—1	—1			—1	1	1			2	—2	—2	9
	3		A_4	2	3	3		1	1					—4	1	1	

←	4	7	A_8	−1/2	1/2	1			1		1/2	1		$\boxed{-1}$		3/2	
→	5	4	A_3	2	1		1							1		−3	
	6	—	0	—										2			
	0	—	Δ	4	1	4			5		2	2			1	4	
←	1	3	A_9	−1	$\boxed{-1}$				1			1	1		1	3	
	2	1	A_6	1		1			1	1	2	2			−2	1	
	3		A_4	3	1	−1		1	−3		−2	−4			1	−5	10
→	4		A_{12}	1/2	−1/2	−1			−1		−1/2	−1		1		−3/2	
	5	4	A_3	3/2	3/2	1	1		1		1/2	1				−3/2	
	6	—	0	—	1												
	0	—	Δ	3		4			6		2	3	1		2	7	
→	1	2	A_1	1	1				−1			−1	−1		−1	−3	
	2	1	A_6	1		1			1	1	2	2			−2	1	11
	3		A_4	2		−1		1	−2		−2	−3	1		2	−2	
	4	4	A_5			1	1		5/2		1/2	5/2	3/2		3/2	3	

effectue ensuite la transformation du tableau. Le résultat est porté dans le tableau 2. Dans la quatrième case de la colonne A_0 du tableau 2 figure un nombre négatif. Par conséquent, la deuxième grande itération n'est pas encore terminée. A l'étape suivante de cette itération, on élimine de la pseudobase le vecteur A_3 et on le remplace par le vecteur supplémentaire A_{10}. Il devient alors possible de condenser le tableau en rayant la ligne 3 et la colonne A_{10} (pour plus de clarté, elles sont mises en relief). La colonne A_0 du tableau 3 ne compte pas d'éléments négatifs, mais comporte des composantes fractionnaires. La deuxième grande itération s'achève par le résultat 3°. Une fois appliquée la première règle de choix de la contrainte supplémentaire, on forme celle-ci d'après la ligne nulle du tableau 3. La grande itération associée à cette contrainte supplémentaire se compose de trois étapes de la méthode duale du simplexe enregistrées dans les tableaux 4 à 6. A la dernière étape on ajoute à la pseudobase le vecteur supplémentaire A_{11}; le tableau 6 est réduit grâce à la suppression de la ligne 1 et de la colonne A_{11}. La quatrième grande itération est associée à une contrainte supplémentaire formée suivant la deuxième ligne du tableau 6 et se compose de deux itérations (tableaux 7 et 8). La dernière (cinquième) grande itération commence par la construction de la contrainte supplémentaire, déterminée par la ligne nulle du tableau 8. L'itération compte trois étapes de la méthode duale du simplexe et s'achève par le résultat 1°. La deuxième étape de l'itération consiste à retrancher le vecteur supplémentaire A_{12}. La solution recherchée X^* du problème (3.29)-(3.32) est définie par les éléments de la colonne A_0 du tableau 11:

$$X^* = (1, 0, 0, 2, 0, 1, 0, 0, 0);$$

de plus

$$L(X^*) = 3.$$

Ainsi, la résolution du problème (3.29)-(3.32) a demandé 11 itérations de la méthode duale du simplexe, sans compter l'étape préalable que nous avons omise. Au cours de la résolution, la valeur de la forme linéaire (3.29) est passée de 8 à 3. L'algorithme a imposé l'introduction de cinq variables additionnelles dont deux seulement, x_{13} et x_{14}, sont restées dans le tableau final. Il s'est trouvé que le programme optimal X^* repose sur une arête du polyèdre des conditions. Il convient de noter qu'une telle situation est plutôt une exception qu'une règle.

Le plus probable est qu'un problème en nombres entiers choisi arbitrairement aura pour solution un point intérieur de l'ensemble convexe des conditions. C'est pourquoi le tableau définitif comprendra le plus souvent $n + 1$ lignes, où n est le nombre de variables du problème.

Le paragraphe suivant, consacré à l'algorithme de programmation partiellement en nombres entiers, établit une nouvelle règle de formation de la contrainte supplémentaire, qui, appliquée aux problèmes totalement en nombres entiers, conduit à des contraintes plus serrées. L'utilisation de cette règle diminue, semble-t-il, dans la plupart des cas le nombre total d'itérations nécessaires (cf. paragraphe 4.4).

Soulignons que la quantité de travail imposé par la résolution des problèmes en nombres entiers dépend essentiellement du choix de la ligne du tableau utilisée pour la construction de la contrainte supplémentaire. Par exemple, le fait de ne pas tenir compte de la ligne nulle du tableau lors du choix de l porterait le processus de résolution du problème (3.29)-(3.32) à 24 itérations.

§ 4. Algorithme de programmation partiellement en nombres entiers

4.1. Considérons le problème de programmation linéaire partiellement en nombres entiers qui consiste à maximiser la forme linéaire

$$L(X) = \sum_{j=1}^{n} c_j x_j \tag{4.1}$$

sous les conditions

$$\sum_{j=1}^{n} a_{ij} x_j = b_i, \quad i = 1, 2, \ldots, m, \tag{4.2}$$

$$x_j \geqslant 0, \quad j = 1, 2, \ldots, n, \tag{4.3}$$

$$x_j \text{ entier pour } j = 1, 2, \ldots, n_1 \leqslant n. \tag{4.4}$$

La méthode de résolution du problème (4.1)-(4.4) est régie par les mêmes principes que l'algorithme du paragraphe précédent. Elle se compose de plusieurs grandes itérations. Chaque itération débute par la formation d'une contrainte additionnelle, en retranchant le programme obtenu plus tôt que ne vérifie pas la restriction (4.4) et qui est en même temps satisfait sur chacun des programmes du problème (4.1)-(4.4).

Rejetons les conditions (4.4) et recherchons la solution du problème de programmation linéaire (4.1)-(4.3).

Supposons que cette solution soit le programme de base $X^{(1)}$, à base $A_{s_1}, A_{s_2}, \ldots, A_{s_m}$, et que les paramètres du tableau optimal soient désignés par x_{ij} ($i = 0, 1, \ldots, m$; $j = 0, 1, \ldots, n$). Si pour tout $s_i \leqslant n_1$ la composante de base x_{0i} est un entier, le programme obtenu est alors solution du problème partiellement en nombres entiers (4.1)-(4.4). Dans le cas contraire, on introduit une contrainte additionnelle dont la règle de construction est établie ci-dessous. Supposons que $s_i \leqslant n_1$ et que, de plus, la partie fractionnaire γ_{0i} du nombre x_{0i} soit positive.

Tout vecteur X qui satisfait aux conditions (4.2) doit vérifier l'égalité

$$x_{s_i} = x_{i0} - \sum_{j \in J} x_{ij} x_j. \tag{4.5}$$

Si la s_i-ième composante de X est un entier, d'après (4.5),

$$\gamma_{i0} - \sum_{j \in J} x_{ij} x_j = n_i \text{ est un entier.} \tag{4.6}$$

Supposons que J_1 (J_2) soit constitué d'indices $j \in J$ tels que x_{ij} est un nombre positif (négatif).

Si

$$\sum_{j\in J} x_{lj}x_j = \sum_{j\in J_1} x_{lj}x_j + \sum_{j\in J_2} x_{lj}x_j \geqslant 0,$$

l'entier n_l de l'égalité

$$\sum_{j\in J_1} x_{lj}x_j + \sum_{j\in J_2} x_{lj}x_j = \gamma_{l0} - n_l, \tag{4.7}$$

déduite de (4.6) ne peut être positif, car s'il l'était, le second membre de (4.7) deviendrait négatif.

On obtient donc, compte tenu de la non-négativité des composantes x_j:

$$\sum_{j\in J_1} x_{lj}x_j \geqslant \sum_{j\in J_1} x_{lj}x_j + \sum_{j\in J_2} x_{lj}x_j \geqslant \gamma_{l0}. \tag{4.8}$$

Mais si

$$\sum_{j\in J} x_{lj}x_j = \sum_{j\in J_1} x_{lj}x_j + \sum_{j\in J_2} x_{lj}x_j < 0,$$

l'entier n_l de l'égalité (4.7) est nécessairement positif, puisque c'est seulement dans ce cas que le second membre de (4.7) devient négatif. Par suite,

$$\sum_{j\in J_2} x_{lj}x_j \leqslant \sum_{j\in J_1} x_{lj}x_j + \sum_{j\in J_2} x_{lj}x_j \leqslant \gamma_{l0} - 1,$$

ou

$$-\frac{\gamma_{l0}}{1-\gamma_{l0}} \sum_{j\in J_2} x_{lj}x_j \geqslant \gamma_{l0}. \tag{4.9}$$

Ainsi, tout programme du X-problème (4.1)-(4.3) ayant une composante entière x_{s_l} vérifie soit (4.8) soit (4.9).

Tenant compte ensuite de la non-négativité des premiers membres des inégalités (4.8) et (4.9), on peut écrire l'inégalité

$$\sum_{j\in J_1} x_{lj}x_j - \frac{\gamma_{l0}}{1-\gamma_{l0}} \sum_{j\in J_2} x_{lj}x_j \geqslant \gamma_{l0}, \tag{4.10}$$

qui est vraie pour tout vecteur X vérifiant les conditions (4.2), (4.3) et comportant une composante entière x_{s_l}.

Nous n'avons pas tenu compte jusqu'à présent des variables x_j (avec $j \in J$) qui satisfont aux conditions (4.4) et qui sont astreintes à être entières. Supposons que $k \leqslant n_1$ corresponde à une variable secondaire, c'est-à-dire $k \in J$.

Dans ce cas, la relation (4.7) reste vraie si l'on substitue à x_{lk} n'importe quel nombre x'_{lk} qui diffère d'un nombre entier de x_{lk}. Après avoir effectué ensuite les raisonnements qui ont déjà permis de déduire l'inégalité (4.10), nous obtenons une condition qui ne

diffère de (4.10) que par le coefficient δ_{lk} de la variable x_k. Dans ce cas,

$$\delta_{lk}=\begin{cases} x'_{lk} & \text{si} \quad x'_{lk}\geqslant 0, \\ -\dfrac{\gamma_{l0}}{1-\gamma_{l0}}x'_{lk} & \text{si} \quad x'_{lk}<0. \end{cases} \tag{4.11}$$

Puisque notre but est de construire la plus forte contrainte possible, il est souhaitable de rendre le coefficient de x_k aussi petit que possible. Si x'_{lk} est choisi non négatif, alors, d'après (4.11), sa valeur optimale sera γ_{lk}, partie fractionnaire de x_{lk}. Par contre, si l'on cherche x'_{lk} parmi les nombres négatifs, il faut pour la même raison poser $x'_{lk}=\gamma_{lk}-1$, qui est le nombre négatif maximal différant de x_k d'un nombre entier.

Ainsi, δ_{lk} peut être rendu égal soit à γ_{lk}, soit à $\frac{\gamma_{l0}}{1-\gamma_{l0}}(1-\gamma_{lk})$. Il est naturel de prendre pour ce coefficient

$$\min\left\{\gamma_{lk}, \frac{\gamma_{l0}}{1-\gamma_{l0}}(1-\gamma_{lk})\right\}=\begin{cases} \gamma_{lk} & \text{si } \gamma_{lk}\leqslant\gamma_{l0}, \\ \dfrac{\gamma_{l0}}{1-\gamma_{l0}}(1-\gamma_{lk}) & \text{si} \quad \gamma_{lk}>\gamma_{l0}. \end{cases}$$

Ces considérations se rapportent à n'importe quelle variable entière x_j avec $j\in J$.

C'est pourquoi la condition (4.10) peut être remplacée par une contrainte plus forte

$$\sum_{j\in J}\delta_{lj}x_j\geqslant\gamma_{l0}, \tag{4.12}$$

où

$$\delta_{lj}=\begin{cases} x_{lj} & \text{si} \quad j>n_1,\ x_{lj}\geqslant 0, \\ -\dfrac{\gamma_{l0}}{1-\gamma_{l0}}x_{lj} & \text{si} \quad j>n_1,\ x_{lj}<0, \\ \gamma_{lj} & \text{si} \quad j\leqslant n_1,\ \gamma_{lj}\leqslant\gamma_{l0}, \\ \dfrac{\gamma_{l0}}{1-\gamma_{l0}}(1-\gamma_{lj}) & \text{si} \quad j\leqslant n_1,\ \gamma_{lj}>\gamma_{l0}. \end{cases} \tag{4.13}$$

L'inégalité (4.12), munie de coefficients δ_{lj} définis par la l-ième ligne du tableau à l'aide de la formule (4.13), est la contrainte additionnelle recherchée.

Une fois écrite sous la forme équivalente

$$-\gamma_{l0}=-\sum_{j\in J}\delta_{lj}x_j+x_{n+1},\quad x_{n+1}\geqslant 0, \tag{4.14}$$

elle s'ajoute aux contraintes linéaires que l'on avait déjà eues. Le programme disponible à base composée de vecteurs A_j avec $j\notin J$ ne satisfait évidemment pas à (4.12), puisqu'il rend nul son premier

membre. Par conséquent, (4.12) ou (4.14) ne se déduit pas des contraintes linéaires antérieures et remplit son rôle « tranchant ». Remarquons que la variable additionnelle x_{n+1}, de même que toutes les variables additionnelles ultérieures, n'est pas bornée par la restriction d'intégralité.

4.2. L'algorithme de programmation partiellement en nombres entiers ne diffère presque en rien de l'algorithme du § 3. Chaque grande itération s'achève par l'un des trois résultats suivants.

1°. Dans la première colonne du dernier tableau de l'itération précédente tous les éléments associés aux variables bornées par la restriction (4.4) sont des entiers; tous les éléments de la colonne sont non négatifs.

2°. La méthode duale du simplexe, dont les itérations (étapes) composent une grande itération, a abouti à un tableau intermédiaire comportant une ligne dont le premier élément est négatif et dont les éléments successifs sont non négatifs.

3°. Dans la première colonne du dernier tableau de l'itération précédente parmi les éléments associés à x_j avec $j \leqslant n_1$ il en est qui sont fractionnaires; tous les éléments de la colonne sont non négatifs.

Les deux premiers résultats marquent la fin de la procédure de résolution: le cas 1° signifie que le vecteur dont les composantes non nulles figurent dans les cases respectives de la première colonne du dernier tableau est un programme optimal du problème (4.1)-(4.4); le cas 2° indique que le problème (4.1)-(4.4) est irréalisable, les contraintes (4.2)-(4.4) étant incompatibles.

Enfin, le cas 3° impose le passage à la grande itération suivante.

A cet effet on choisit la ligne l du dernier tableau de l'itération précédente qui correspond à une variable bornée par la restriction (4.4) et contient dans la première case un nombre fractionnaire x_{0l}. Le choix d'une telle ligne peut se faire en utilisant l'une des règles du paragraphe 3.6. La contrainte additionnelle (4.13), (4.14) se forme d'après la l-ième ligne du tableau. Le tableau résultant est bordé d'une ligne composée d'éléments $-\delta_{lj}$ et d'une colonne unité à unité figurant dans la dernière case, cette colonne correspondant à la nouvelle variable additionnelle. Ensuite viennent plusieurs étapes de la méthode duale du simplexe, qui amènent l'un des trois résultats énumérés ci-dessus. Si à l'une des étapes une variable additionnelle quelconque devient variable de base, la ligne et la colonne correspondantes du tableau sont rayées.

Le caractère fini de la méthode de programmation partiellement en nombres entiers est garanti lorsque sont respectées les conditions du théorème 3.1 ainsi qu'une restriction supplémentaire d'après laquelle la forme linéaire (4.1) ne doit dépendre que des n_1 premières variables vérifiées par les contraintes d'intégralité (4.4). Cette dernière hypothèse doit assurer l'intégralité des valeurs de la forme

linéaire (4.1) sur tous les programmes partiellement en nombres entiers du problème.

La démonstration de la finitude de la méthode dans ces conditions peut s'effectuer à l'aide des raisonnements utilisés aux paragraphes 3.4 et 3.5 pour établir que la méthode du § 3 est finie. Après avoir acquis la connaissance de ces paragraphes, le lecteur pourra effectuer cette démonstration lui-même (cf. exercice 9). Si la forme linéaire (4.1) est définie par certaines variables non assujetties aux restrictions (4.4), la finitude de la méthode ne peut pas être prouvée. Le fait est que dans ce cas la ligne associée à la variable $x_0 = L(X)$ ne participe pas à la formation des contraintes additionnelles. Il se peut donc qu'il se forme une suite infinie $X^{(\mu)}$ qui donne lieu aux relations

$$\lim_{\mu\to\infty} L(X^{(\mu)}) = L^*, \quad L(X^{(\mu)}) > L^*.$$

Il est toutefois à présumer qu'une telle éventualité se présente rarement et que l'algorithme permet généralement d'obtenir la solution du problème pour un type quelconque de forme linéaire (4.1).

4.3. Considérons à titre d'exemple le problème (3.29)-(3.31) soumis aux conditions supplémentaires

$$x_j \text{ entier pour } j = 1, 2, 3, 4, 5, 9. \tag{4.15}$$

Ce problème se distingue de (3.29)-(3.32) du paragraphe 3.7 par les variables x_6, x_7 et x_8 qui ne sont pas supposées entières. La procédure de résolution de ce problème partiellement en nombres entiers est traduite par les tableaux 5.2. Dans le tableau 0 figurent les paramètres du programme optimal du problème (3.29)-(3.31) à base (A_8, A_9) (le résultat de l'étape préalable est ici omis). La ligne 2 de la colonne A_0 de ce tableau comporte un nombre fractionnaire, ce qui signifie que le programme disponible n'obéit pas aux restrictions (4.15). La première grande itération commence par la formation de la contrainte additionnelle. Ici, le choix de la ligne qui engendre cette contrainte est univoque: c'est la ligne 2 du tableau 0. Ajoutons au tableau 0 une ligne (la ligne 3) et une colonne $A_{10}=e_4$. Portons $-\frac{5}{18} = -\left(\frac{41}{18} - \left[\frac{41}{18}\right]\right)$ dans la case 0 de la ligne 3. Les éléments des cases 1 à 9 de la ligne 3 se calculent d'après la formule (4.13), où $l = 2$. Ensuite, on complète la ligne θ et l'on choisit la colonne directrice A_3. La ligne 3 et la colonne A_3 déterminent la transformation du tableau, transformation dont le résultat est inscrit dans les quatre premières lignes du tableau 1. Dans la deuxième case de la colonne A_0 du tableau 1 figure un nombre négatif, ce qui veut dire que la première grande itération n'est pas terminée. Après avoir effectué encore une étape de la méthode duale du simplexe, on aboutit au tableau 2 dont la colonne A_0 a tous ses éléments non négatifs, mais $x_{30}^{(2)} = x_3^{(2)} = \frac{3}{8}$ est un nombre fractionnaire. Par conséquent, la grande itération s'est achevée par le résultat 3°. Le choix de la ligne qui engendre la nouvelle contrainte additionnelle est de nouveau univoque: ce ne peut être que la ligne 3. On forme cette contrainte en augmentant le tableau 2 d'une ligne et d'une colonne. La nouvelle transformation du tableau conduit au tableau 3. Les éléments de la colonne A_0 du tableau 3 sont non négatifs, et de plus, les seuls éléments qui apparaissent comme fractionnaires sont ceux des 1-ère et

Tableau 5.2

										Non entiers						
		c_j	—	—	2	1	4		3	1	2	7	3			—
	N	C_X	B_X	A_0	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}	A_{11}	Nombre d'itér.
	0	—	Δ	8	2	4	0	1	3	1	2					
	1	7	A_8	0,167	0,584	0,584	0,500	—0,167	0,917	—0,083	0,417	1				
+	2	3	A_9	2,278	—0,028	0,356	0,167	0,722	—0,139	0,861	0,361		1			0
←	3		A_{10}	—0,278	—0,011	—0,267	$\boxed{-0,167}$	—0,107	—0,053	—0,861	—0,361			1		
	4	—	θ	—	187,2	14,98		9,36	56,16	1,161	5,538					
	0	—	Δ	8	2	4	0	1	3	1	2					
←	1	7	A_8	—0,667	0,551	—0,218		—0,487	0,756	$\boxed{-2,167}$	—0,167	1		3		
	2	3	A_9	2	—0,038	0,038		0,615	—0,192				1	1		1

→	3	4	A_3	1,667	0,064	1,603	1	0,641	0,321	5,167	2,167			—6		
	4	—	θ	—		18,35		2,053		0,375	3					
	0	—	Δ	7,75	2,207	3,918		0,817	3,284		1,75	0,375		1,125		
→	1	1	A_6	0,25	—0,207	0,082		0,183	—0,286	1	0,25	—0,375		—1,125		
	2	3	A_9	2	—0,038	0,038		0,615	—0,192				1	1		2
+	3	4	A_3	0,375	1,132	1,180	1	—0,303	1,786		0,875	1,938		—0,188		
←	4		A_{11}	—0,375	—0,132	—0,180		—0,182	—0,128		—0,875	$\boxed{-1,938}$		—0,112	1	
	5	—	0	—	16,69	21,73		4,497	25,58		2	0,194		10		
	0	—	Δ	7,677	2,181	3,883		0,782	3,259		1,581			1,103	0,194	
	1	1	A_6	0,323	—0,181	0,118		0,218	—0,259	1	0,419			—1,103	—0,194	
	2	3	A_9	2	—0,038	0,038		0,615	—0,192				1	1		3
	3	4	A_3	0	1	1	1	—0,485	1,658					—0,300	1	
→	4	7	A_8	0,194	0,068	0,093		0,094	0,066		0,451	1		0,058	—0,516	

Tableau 53.

		c_j	—	—	2	1	4		3	1	2	7	3				—
	N	C_X	B_X	A_0	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}	A_{11}	A_{12}	Nombre d'itér.
	0	—	Δ	8	2	4		1	3	1	2						
+	1	7	A_8	1/6	7/12	7/12	1/2	—1/6	11/12	—1/12	5/12	1					
	2	3	A_9	41/18	—1/36	11/36	1/6	13/18	—5/36	31/36	13/36		1				0
←	3		A_{10}	—1/6	—1/12	—1/12	$\boxed{-1/10}$	—1/30	—1/60	—1/60	—7/60			1			
	4	—	θ	—	24	48	0	30	180	60	120/7						
	0	—	Δ	8	2	4		1	3	1	2						
←	1	7	A_8	—2/3	1/6	1/6		$\boxed{-1/3}$	5/6	—1/6	—1/6	1		5			
	2	3	A_9	2	—1/6	1/6		2/3	—1/6	5/6	1/6		1	5/3			1
→	3	4	A_3	5/3	5/6	5/6	1	1/3	1/6	1/6	7/6			—10			
	4	—	θ	—				3		6	12						
	0	—	Δ	6	5/2	9/2			11/2	1/2	3/2	3		15			
→	1		A_4	2	—1/2	—1/2		1	—5/2	1/2	1/2	—3		—15			
+	2	3	A_9	2/3	1/6	1/2			3/2	1/2	—1/6	2	1	35/3			2

	3	4	A_3	1	1	1	1		1		1	1		−5			
←	4		A_{11}	−2/3	−1/6	−1/2			−1/2	$\boxed{-1/2}$	−1/3			−2/3	1		
	5	—	0	—	15	9			11	1	9/2			45/2			
+	0	—	Δ	16/3	7/3	4			5		7/6	3		43/3	1		
	1		A_4	4/3	−2/3	−1		1	−3		1/6	−3		−47/3	1		
	2	3	A_9	0					1		−1/2	2	1	11	1		
	3	4	A_3	1	1	1	1		1		1	1		−5			3
→	4	1	A_6	4/3	1/3	1			1	1	2/3			4/3	−2		
←	5		A_{12}	−1/3	$\boxed{-1/3}$						−1/6			−1/3		1	
	6	—	0	—	7						7			43			
	0	—	Δ	3		4			5			3		12	1	7	
	1		A_4	2		−1		1	−3		1/2	−3		−15	1	−2	
	2	3	A_9	0					1		−1/2	2	1	11	1		
	3	4	A_3	0		1	1		1		1/2	1		−6		3	4
	4	1	A_6	1		1			1	1	1/2			1	−2	1	
→	5	2	A_1	1	1						1/2			1		−3	

4-ième cases associées respectivement aux variables x_6 et x_8. Par conséquent, la grande itération, qui comprend deux étapes, s'achève par le résultat 1°. La solution du problème est définie par le vecteur

$$X^* = (0, 0, 0, 0, 0, 10/31, 0, 6/31, 2).$$

Ainsi, la procédure de résolution du problème se compose de trois étapes de la méthode duale du simplexe.

4.4. La règle de formation de la contrainte additionnelle traduite par la formule (4.13) peut être appliquée également aux problèmes totalement en nombres entiers ($n_1 = n$) examinés au paragraphe précédent. La formule (4.13) prend la forme

$$\delta_{ij} = \begin{cases} \gamma_{ij} & \text{si} \quad \gamma_{ij} \leqslant \gamma_{i0}, \\ \dfrac{\gamma_{i0}}{1-\gamma_{i0}}(1-\gamma_{ij}) & \text{si} \quad \gamma_{ij} > \gamma_{i0}. \end{cases} \tag{4.16}$$

Au lieu de la contrainte additionnelle

$$\gamma_{i0} \leqslant \sum_{j \in J} \gamma_{ij} x_j, \tag{4.17}$$

qui est à la base de l'algorithme du § 3, introduisons une contrainte additionnelle nouvelle

$$\gamma_{i0} \leqslant \sum_{j \in J} \delta_{ij} x_j, \tag{4.18}$$

dont les nombres δ_{ij} se calculent d'après la formule (4.16).

Si tous les γ_{ij} ne dépassent pas γ_{i0}, (4.18) coïncide avec (4.17). Mais si certains des γ_{ij} sont plus grands que γ_{i0}, les coefficients des contraintes (4.18) qui leur répondent sont strictement inférieurs aux coefficients correspondants de (4.17) :

$$\gamma_{ij} - \delta_{ij} = \frac{\gamma_{ij} - \gamma_{i0}}{1-\gamma_{i0}} > 0.$$

Par conséquent, la contrainte (4.18) n'est jamais plus faible que la condition (4.17) et dans de nombreux cas elle est même plus forte. L'utilisation de contraintes additionnelles du type (4.18) diminue généralement le nombre d'étapes nécessaires pour résoudre le problème.

Résolvons à titre d'exemple le problème (3.29)-(3.32), examiné au paragraphe 3.7, en utilisant la nouvelle règle de formation des contraintes additionnelles.

La procédure correspondante est donnée par le tableau 5.3. La solution s'obtient en quatre itérations qui forment deux grandes itérations. Rappelons que ce même exemple fut résolu au paragraphe 3.7 avec emploi de contraintes additionnelles du type (4.17), ceci en 11 itérations. Ainsi dans le cas considéré la nouvelle règle s'est avérée très efficace.

§ 5. Autres méthodes de programmation en nombres entiers

5.1. Voici une brève description d'une autre méthode de résolution des problèmes en nombres entiers [26]. Cette méthode, de même que les algorithmes du paragraphe précédent, est fondée sur la troncature successive du polyèdre convexe des conditions du problème et s'apparente par sa technique de calcul à la méthode duale du simplexe. Pourtant, à la différence de l'algorithme du § 3, elle nécessite à chaque étape l'introduction d'une contrainte additionnelle, et il se trouve de plus que les paramètres x_{ij} de tout tableau intermédiaire sont des entiers. Revenons au problème (3.1)-(3.4) et supposons que l'on ait trouvé un certain système de ses vecteurs des conditions $A_{s_1}, \ldots, A_{s_m}$ qui jouit des propriétés suivantes:

les coefficients de décomposition x_{ij} de n'importe quel vecteur A_j ($j = 0, 1, \ldots, n$) sont des entiers;

les critères x_{0j} de tous les vecteurs des conditions sont non négatifs.

Si chacun des nombres x_{i0} ($i = 1, 2, \ldots, m$) est non négatif, le programme à base $A_{s_1}, A_{s_2}, \ldots, A_{s_m}$ est la solution cherchée du problème.

Supposons que parmi les x_{i0} il y ait des nombres négatifs. Cela signifie que le système de vecteurs $A_{s_1}, A_{s_2}, \ldots, A_{s_m}$ est une pseudobase (base d'un certain pseudoprogramme).

On pourrait certes choisir l'un des indices i tel que $x_{i0} < 0$ et effectuer une itération de la méthode duale du simplexe en faisant décroître la valeur de la forme linéaire du problème. Toutefois, dans ce cas, la condition d'intégralité des paramètres du tableau serait en règle générale enfreinte à cause de la nécessité de diviser par l'élément directeur de la transformation. On ne peut garantir que les paramètres du nouveau tableau soient en nombres entiers que lorsque l'élément directeur de la transformation est égal à —1.

Or, n'est-il pas possible de former une nouvelle contrainte à laquelle satisfassent tous les programmes entiers et qui cependant détermine la ligne génératrice de la transformation dont l'élément directeur est —1? Il se trouve que cela est possible. Supposons que $x_{l0} < 0$ et il y ait parmi les x_{lj} des nombres négatifs. La contrainte additionnelle se construit d'après la condition

$$x_{s_l} = x_{l0} - \sum_{j \in J} x_{lj} x_j \tag{5.1}$$

associée à la l-ième ligne du tableau ($l = 1, 2, \ldots, m$). Choisissons un nombre arbitraire $\alpha > 0$ et considérons les rapports

$$\frac{x_{lj}}{\alpha},$$

qu'il est commode de mettre sous la forme

$$\frac{x_{lj}}{\alpha}=\left[\frac{x_{lj}}{\alpha}\right]+\left(\frac{x_{lj}}{\alpha}-\left[\frac{x_{lj}}{\alpha}\right]\right).$$

En désignant par $n_{lj}(\alpha)$ et $\gamma_{lj}(\alpha)$ respectivement les parties entière et fractionnaire de $\frac{x_{lj}}{\alpha}$, on obtient

$$\frac{x_{lj}}{\alpha}=n_{lj}(\alpha)+\gamma_{lj}(\alpha), \quad 0\leqslant\gamma_{lj}<1. \tag{5.2}$$

Divisons (5.1) par α et transformons l'égalité obtenue à l'aide de (5.2) :

$$\frac{x_{s_l}}{\alpha}+\sum_{j\in J}\gamma_{lj}(\alpha)\,x_j-\gamma_{l0}(\alpha)=n_{l0}(\alpha)-\sum_{j\in J}n_{lj}(\alpha)\,x_j. \tag{5.3}$$

Avec $x_j\geqslant 0$, le premier membre de (5.3) est la différence entre une quantité non négative et $\gamma_{l0}(\alpha)<1$ et est donc strictement supérieur à -1. Par conséquent, le deuxième membre de (5.3), étant entier avec x_j entiers, vérifie la condition

$$n_{l0}(\alpha)-\sum_{j\in J}n_{lj}(\alpha)\,x_j\geqslant 0, \tag{5.4}$$

vraie pour tout programme du problème (3.1)-(3.4). Introduisant une nouvelle variable x_{n+1}, nous récrivons (5.4) sous la forme

$$x_{n+1}=n_{l0}(\alpha)-\sum_{j\in J}n_{lj}(\alpha)\,x_j, \quad x_{n+1}\geqslant 0. \tag{5.5}$$

D'après (5.2), la négativité de x_{lj} entraîne celle de $n_{lj}(\alpha)$, et inversement, la négativité de $n_{lj}(\alpha)$ entraîne celle de x_{lj}.

Par suite, $n_{l0}<0$ et parmi les nombres n_{lj} il en est qui sont négatifs, ce qui signifie que la ligne du tableau définie par la nouvelle contrainte peut être prise comme ligne génératrice pour la transformation ultérieure. De plus, avec

$$\alpha\geqslant\max_{j\in J}(-x_{lj}),$$

à chaque x_{lj} négatif, $j\in J$, correspond $n_{lj}(\alpha)=-1$ [cf. (5.2)], c'est-à-dire l'élément directeur de cette transformation est d'avance égal à -1.

5.2. Voici les considérations qui permettent de guider le choix du nombre α qui entre dans (5.5). La forme linéaire (3.1), à la suite de la transformation de l'élément directeur $n_{lk}=-1$, décroît d'une quantité

$$-n_{l0}\cdot x_{0k}.$$

Comme on le voit, la décroissance de la forme linéaire ne peut que diminuer avec la croissance de α. C'est pourquoi il est rationnel de prendre pour α le plus petit nombre possible α_0 tel que l'élément directeur soit égal à -1. Il est clair que

$$\alpha_0 \leqslant \max_{j \in J} (-x_{lj})$$

On peut déterminer la valeur exacte de α_0 en se guidant sur la règle suivante. Supposons l'indice k choisi d'après la condition

$$x_{0k} = \min_{x_{lj} < 0} x_{0j}, \quad x_{lk} < 0. \tag{5.6}$$

Sélectionnons pour tout $x_{lj} < 0$ le plus grand nombre naturel possible d_j tel que

$$\frac{x_{0j}}{d_j} \geqslant x_{0k}. \tag{5.7}$$

Dans ce cas,

$$\alpha_0 = \max_{x_{lj} < 0} \left(-\frac{x_{lj}}{d_j}\right). \tag{5.8}$$

Nous laissons au lecteur le soin de justifier cette règle (cf. exercice 12).

Après avoir calculé α_0, posons $\alpha = \alpha_0$ dans (5.5) et ajoutons au tableau disponible la colonne unité e_{m+1} et la ligne aux éléments $n_{lj}(\alpha_0)$ ($n_{lj}(\alpha) = 0$ si $j = s_1, s_2, \ldots, s_m$).

Transformons ensuite le tableau en prenant pour ligne directrice la nouvelle ($(m+1)$-ème) ligne et en choisissant la colonne directrice d'après les règles de la méthode duale du simplexe. Le choix de α_0 égalise l'élément directeur de la transformation à -1. Par conséquent, le nouveau tableau jouit de la propriété du tableau précédent, c'est-à-dire se compose de nombres entiers x'_{ij}, et de plus, $x'_{0j} \geqslant 0$ pour $j = 1, 2, \ldots, n+1$. La suite d'opérations ainsi décrite constitue une itération isolée de l'algorithme. Les itérations sont effectuées jusqu'à ce que l'on obtienne un tableau dans lequel ou bien toutes les composantes de base seront non négatives, ou bien une ligne aura son premier élément négatif et ses autres éléments non négatifs. Dans le premier cas, le problème est résolu; dans le deuxième, ses conditions sont contradictoires.

Si, à l'une des étapes, une variable additionnelle quelconque étant d'abord secondaire devient ensuite variable de base, la ligne et la colonne qui lui correspondent sont rayées. Il est bon de faire le choix de la ligne l qui engendre la contrainte additionnelle en se guidant sur la condition de décroissance maximale de la fonction (3.1) à l'étape considérée.

Cette méthode suppose l'existence d'un tableau initial aux nombres entiers et aux valeurs non négatives des critères. La construc-

tion d'un tel tableau ne demande quelquefois pas de calculs. Mais, dans le cas général, l'étape préalable se ramène à l'application de la même méthode à un certain problème auxiliaire; de plus, l'une des itérations de cette méthode peut mettre en évidence l'impossibilité de construire le tableau initial, ce qui indique naturellement que le problème initial n'a pas de solution.

Compte tenu de la complexité de l'étape préalable, il est douteux qu'on puisse appliquer raisonnablement la méthode décrite aux problèmes dans lesquels le contenu du tableau initial n'est pas évident.

Si, à chaque étape de la méthode, la colonne directrice est choisie d'après les critères vectorielles $\vec{x}_{0j}$ (cf. paragraphe 3.4), et si toutes les N itérations (N étant un nombre fixé) la ligne l est choisie à partir de la condition

$$s_l = \min_{x_{i0}<0} s_i, \quad x_{l0} < 0,$$

alors, dans le cas où la forme linéaire (3.1) est bornée inférieurement sur l'ensemble (3.2)-(3.3), la finitude de la méthode est assurée.

5.3. Le lecteur pourrait avoir l'impression que toute règle de formation des contraintes additionnelles, susceptible d'assurer la troncature du programme non entier disponible tout en conservant tous les programmes entiers, conduit à la méthode de programmation en nombres entiers. Le but du présent paragraphe est de montrer qu'un tel raisonnement est erroné et de prévenir le lecteur de la nécessité d'observer une certaine prudence en utilisant les différentes règles de troncature des polyèdres.

Considérons encore une fois le problème en nombres entiers (3.1)-(3.4). Supposons que $X^{(1)}$ soit une solution du problème (3.1)-(3.3), dont certaines composantes sont fractionnaires. Introduisons une contrainte additionnelle du type

$$\sum_{j \in J} x_j \geqslant 1, \tag{5.9}$$

où J est l'ensemble des indices des variables secondaires du programme $X^{(1)}$.

Le programme $X^{(1)}$ ne satisfait pas à (5.9), puisqu'il rend nul le premier membre de cette relation.

D'autre part, si X est un programme entier quelconque, alors parmi ses composantes x_j, $j \in J$, il en est qui sont positives, et le premier membre de (5.9) est égal à un entier positif, ce qui signifie que la condition (5.9) est observée. Ainsi, la contrainte (5.9), proposée en 1959 par Dantzig [36], jouit en effet de deux propriétés caractéristiques des conditions supplémentaires de la programmation en nombres entiers. Pourtant, comme l'ont montré Gomory et Hoffman [30], l'algorithme de programmation en nombres entiers,

fondé sur la règle (5.9), ne conduit généralement pas au but. Reprenons les raisonnements bien simples de [30].

Convenons de désigner par $x_{n+\alpha} \geqslant 0$ une variable additionnelle qui transforme en égalité l'α-ème contrainte additionnelle. Considérons un programme arbitraire $X = (x_1, x_2, \ldots, x_n)$ du problème (3.1)-(3.3). Les valeurs des variables additionnelles $x_{n+\alpha}$ sont déterminées univoquement par le programme X, soit $x_{n+\alpha} = x_{n+\alpha}(X)$.

Numérotons les composantes du programme X dans l'ordre de leur croissance et désignons par $\tilde{x}_j(X)$ la valeur de la nouvelle j-ième composante. Ainsi,

$$\tilde{x}_1(X) \leqslant \tilde{x}_2(X) \leqslant \ldots \leqslant \tilde{x}_n(X). \tag{5.10}$$

Posons

$$\sigma(X) = \sum_{j=1}^{n-m-1} \tilde{x}_j(X). \tag{5.11}$$

Lemme 5.1. *Si pour X quelconque*

$$\sigma(X) \geqslant 1, \tag{5.12}$$

alors

$$\tilde{x}_{n+\alpha}(X) \geqslant \tilde{x}_{n-m-1}(X), \quad \alpha = 1, 2, \ldots \tag{5.13}$$

Démonstration. Par définition,

$$x_{n+1}(X) = \sum_{j \in J} x_j - 1.$$

L'ensemble des indices J comprend $n - m$ éléments. Par conséquent, pour un certain $k \in J$ la composante

$$x_k \geqslant \tilde{x}_{n-m-1}(X).$$

La relation évidente

$$\sum_{j \in J} x_j = x_k + \sum_{\substack{j \in J \\ j \neq k}} x_j \geqslant x_k + \sigma(X)$$

et la condition (5.12) entraînent

$$x_{n+1}(X) \geqslant x_k + \sigma(X) - 1 \geqslant \tilde{x}_{n-m-1}(X). \tag{5.14}$$

L'inégalité (5.14) signifie que dans le problème obtenu à partir de (3.1)-(3.3) par addition de la condition (5.9) les variables $\tilde{x}_j(X)$, $j = 1, 2, \ldots, n - m - 1$, gardent leurs valeurs antérieures. En reprenant donc les raisonnements ci-dessus, nous pouvons justifier (5.13) avec $\alpha = 2$. D'une manière analogue on vérifie l'inégalité (5.13) pour α quelconque. Le lemme est ainsi démontré.

Le lemme 5.1 sert de base à la démonstration de la proposition suivante.

Théorème 5.1. *Si chaque programme optimal du problème en nombres entiers* (3.1)-(3.4) *compte au moins* $m+2$ *composantes positives, l'algorithme établi à partir des contraintes additionnelles du type* (5.9) *ne peut aboutir à la solution recherchée.*

Démonstration. Supposons que la proposition du théorème soit fausse, c'est-à-dire que l'algorithme aboutisse en quelques itérations à un programme entier X^* qui est alors, évidemment, solution du problème. Complétons X^* par les variables $x_{n+\alpha}(X^*)$, $\alpha = 1, 2, \ldots, t$ (t est le nombre de contraintes additionnelles introduites).

Parmi les nombres

$$x_1^*,\ x_2^*,\ \ldots,\ x_n^*,\ x_{n+1}(X^*),\ \ldots,\ x_{n+t}(X^*), \tag{5.15}$$

il doit y avoir $n-m$ nombres nuls (variables secondaires dont le nombre reste invariable à chaque étape). Mais, d'après la condition, parmi les entiers x_j^* il y a au moins $m+2$ nombres positifs. Par conséquent,

$$\sigma(X^*) = \sum_{j=1}^{n-m-1} \tilde{x}_j(X^*) \geqslant \tilde{x}_{n-m-1}(X^*) \geqslant 1,$$

et d'après le lemme 5.1

$$x_{n+\alpha}(X^*) \geqslant \tilde{x}_{n-m-1}(X^*) \geqslant 1. \tag{5.16}$$

La relation (5.16) traduit le fait que seuls les n premiers nombres du système (5.15) peuvent être nuls. Ainsi, parmi les composantes de la solution X^* il y a $n-m$ zéros, c'est-à-dire que X^* ne compte pas plus de m composantes positives. La contradiction ainsi obtenue justifie la proposition du théorème.

Le théorème 5.1 montre que la convergence de l'algorithme considéré ne peut avoir lieu que si la solution recherchée se situe sur une arête du polyèdre convexe (3.2)-(3.3) [le problème (3.1)-(3.3) est supposé non dégénéré]. Mais même sous ces hypothèses suffisamment restrictives la convergence de la méthode n'est pas toujours garantie puisque le théorème 5.2 se vérifie. *Si pour la solution* X^* *du problème* (3.1)-(3.4) *et pour un certain programme* X *du problème* (3.1)-(3.3) *est remplie la condition*

$$\sum_{j=1}^{n} c_j x_j > \sum_{j=1}^{n} c_j x_j^*, \tag{5.17}$$

et si de plus

$$\sigma(X) \geqslant 1, \tag{5.18}$$

on sait d'avance que l'algorithme ne converge pas.

Démonstration. Si la proposition du théorème était incorrecte, en vertu de (5.17) à l'une des itérations de l'algorithme le point X serait tronqué, c'est-à-dire pour une certaine variable additionnelle $x_{n+\alpha}$ la valeur de $x_{n+\alpha}(X)$ devrait être négative.

Or, (5.18) et le lemme 5.1 amènent

$$x_{n+\alpha}(X) \geqslant \tilde{x}_{n-m-1}(X) > 0.$$

La contradiction obtenue justifie le théorème.

En résumant les résultats exposés, on peut constater que la convergence d'un algorithme fondé sur les contraintes additionnelles (5.9), si même elle a lieu, ne peut intéresser qu'une classe de problèmes excessivement étroite.

C'est pourquoi cet algorithme ne peut être recommandé pour les calculs pratiques.

5.4. La solution de certains problèmes de programmation en nombres entiers peut s'obtenir à l'aide de la programmation dynamique (cf. [5]).

Considérons à titre d'exemple le problème du sac à dos (cf. paragraphe 1.3), qui consiste à maximiser la forme linéaire

$$\sum_{j=1}^{n} c_j x_j \tag{5.19}$$

sous les conditions

$$\sum_{j=1}^{n} a_j x_j \leqslant b, \tag{5.20}$$

$$0 \leqslant x_j \leqslant d_j, \quad j = 1, 2, \ldots, n, \tag{5.21}$$

$$x_j \text{ entier}, \quad j = 1, 2, \ldots, n. \tag{5.22}$$

Les paramètres c_j, $a_j > 0$, $b > 0$ et $d_j > 0$ sont ici des entiers.

Introduisons les fonctions $f_k(z)$ de l'argument entier z pour $k = 1, 2, \ldots, n$, en posant $f_k(z)$ égal à la valeur optimale de la forme linéaire du problème (5.19)-(5.22) dans lequel l'indice n et le paramètre b sont remplacés respectivement par k et z.

Les valeurs des fonctions f_k et f_{k-1} sont associées par des relations récurrentes

$$f_k(z) = \max\,[f_{k-1}(z - a_k \cdot h) + c_k \cdot h], \tag{5.23}$$

où le maximum est pris sur les h entiers qui vérifient les conditions

$$0 \leqslant h \leqslant h(z, k) = \min\left\{\left[\frac{z}{a_k}\right], d_k\right\};$$

nous laissons au lecteur la justification de cette relation (cf. exercice 15).

La relation (5.23) permet de calculer successivement les valeurs de toutes les fonctions $f_k(z)$ pour $0 \leqslant z \leqslant b$, si l'on tient compte que

$$f_1(z) = c_1 \min \left\{ \left[\frac{z}{a_1} \right], d_1 \right\}.$$

Pour définir à l'aide de (5.23) la valeur de la fonction f_k au point z, il faut effectuer une opération de division, $h(k, z)$ opérations d'addition et $h(k, z)$ opérations de comparaison. Par conséquent, le calcul de $f_k(z)$ pour $0 \leqslant z \leqslant b$ impose

$$2 \sum_{z=1}^{b} h(k, z) + b$$

opérations. Si l'on suppose que $a_j = 1$, $d_j = d < b$, $j = 1, 2, \ldots, n$, le nombre total d'opérations nécessaires pour définir toutes les fonctions $f_j(z)$, $j = 2, 3, \ldots, n$, s'exprime par la formule

$$d(2b - d + 1)(n - 1). \tag{5.24}$$

Connaissant les fonctions $f_j(z)$, il est aisé de calculer les composantes du programme optimal cherché du X^*-problème (5.19)-(5.22).

Prenons

$$0 \leqslant x_n^* \leqslant d_n \text{ à partir de la condition}$$
$$c_n x_n^* + f_{n-1}(b - a_n x_n^*) = f_n(b);$$
$$0 \leqslant x_{n-1}^* \leqslant d_{n-1} \text{ à partir de la condition}$$
$$c_{n-1} x_{n-1}^* + f_{n-2}(b - a_n x_n^* - a_{n-1} x_{n-1}^*) = f_{n-1}(b - a_n x_n^*);$$

etc.

$$0 \leqslant x_2^* \leqslant d_2 \text{ à partir de la condition}$$
$$c_2 x_2^* + f_1(b - a_n x_n^* - \ldots - a_2 x_n^*) =$$
$$= f_2(b - a_n x_n^* - \ldots - a_3 x_3^*);$$
$$0 \leqslant x_1^* \leqslant d_1 \text{ à partir de la condition}$$
$$c_1 x_1^* = f_1(b - a_n x_n^* - \ldots - a_2 x_2^*).$$

En additionnant les n égalités précédentes, on obtient

$$\sum_{j=1}^{n} c_j x_j^* = f_n(b),$$

ce qui veut dire que le vecteur X^* aux composantes entières x_j^* est solution du problème (5.19)-(5.22).

Les opérations liées à la détermination des composantes x_j^* d'après les fonctions connues $f_j(z)$ s'avèrent superflues si l'on retient chaque fois, lors du calcul du second membre de (5.23), la valeur de l'argument h pour lequel est atteint le maximum.

Si dans le problème (5.19)-(5.22) chacune des variables ne peut prendre que deux valeurs: 0 ou 1, c'est-à-dire si $d_j = 1$ et

$a_j = 1$ pour $1 \leqslant j \leqslant n$, alors, d'après (5.24), sa solution nécessite $2b\,(n-1)$ opérations d'addition et de comparaison. La résolution de ce problème par le traitement de toutes les variantes (leur nombre est C_n^b si $c_j > 0$, $1 \leqslant j \leqslant n$, $b < n$) demande $b \cdot C_n^b$ opérations. Lorsque le nombre n de variables du problème est grand et $b \gg 1$, $b \ll n$, le dernier moyen ne convient manifestement pas, alors que le premier conduit à la solution cherchée par une voie assez économique.

Le mérite du procédé décrit consiste encore en ceci qu'il est peu sensible au type de fonction à maximiser: le travail nécessaire pour la résolution du problème ne s'accroît presque pas si l'on remplace la forme linéaire (5.19) par une fonction non linéaire du type

$$\sum_{j=1}^{n} \varphi_j(x_j),$$

où $\varphi_j(x_j)$ est une fonction arbitraire d'un argument entier.

Toutefois, lorsque le nombre de contraintes du problème en nombres entiers augmente, la quantité de travail à fournir en utilisant la méthode de programmation dynamique s'accroît brusquement. C'est que pour un problème qui compte m contraintes associant toutes les variables, les fonctions f_j sont définies par m variables:

$f_j = f_j(z_1, z_2, \ldots, z_m)$ est la valeur optimale de la forme linéaire d'un problème à j variables dont les seconds membres des contraintes sont égaux à $z_1, z_2, \ldots, z_m$.

C'est pourquoi le volume de calcul des fonctions f_j s'accroît au rythme d'une progression géométrique à mesure que m augmente. Il semble que la limite raisonnable de l'application de la méthode de la programmation dynamique à la résolution du problème en nombres entiers général soit limitée par trois conditions.

La méthode de la programmation dynamique est efficace également pour une série de problèmes en nombres entiers spéciaux.

Considérons, par exemple, le problème de distribution en nombres entiers, qui consiste à maximiser la forme linéaire

$$\sum_{i=1}^{2} \sum_{j=1}^{n} c_{ij} x_{ij} \tag{5.25}$$

sous les conditions

$$\sum_{j=1}^{n} \lambda_{ij} x_{ij} \leqslant a_i, \quad i = 1, 2, \tag{5.26}$$

$$\sum_{i=1}^{2} x_{ij} = b_j, \quad j = 1, 2, \ldots, n, \tag{5.27}$$

$$0 \leqslant x_{ij} \leqslant d_{ij}, \tag{5.28}$$

$$x_{ij} \text{ entier.} \tag{5.29}$$

Ici $c_{ij} \geqslant 0$, $\lambda_{ij} \geqslant 0$.

Introduisons les fonctions

$$F_j(z_1, z_2), \quad 0 \leqslant z_1 \leqslant a_1, \quad 0 \leqslant z_2 \leqslant a_2; \quad j = 1, 2, \ldots, n,$$

en posant $F_k(z_1, z_2)$ égal à la valeur optimale de la forme linéaire du problème (5.25)-(5.29) dans lequel $n = k$, $a_1 = z_1$, $a_2 = z_2$. Les fonctions $F(z_1, z_2)$ se calculent d'après la relation de récurrence

$$F_k(z_1, z_2) =$$

$$= \max [F_{k-1}(z_1 - \lambda_{1k}h_1, z_2 - \lambda_{2k}h_2) + c_{1k}h_1 + c_{2k}h_2], \tag{5.30}$$

dans laquelle le maximum est recherché par rapport aux nombres entiers h_1 et h_2 limités par les conditions

$$\left.\begin{aligned} &0 \leqslant h_1 \leqslant \min\left\{d_{1k}, \left[\frac{z_1}{\lambda_{1k}}\right]\right\}, \\ &0 \leqslant h_2 \leqslant \min\left\{d_{2k}, \left[\frac{z_2}{\lambda_{2k}}\right]\right\}, \quad h_1 + h_2 \leqslant b_k. \end{aligned}\right\} \tag{5.31}$$

Supposons que le maximum de l'expression (5.30) s'obtient avec

$$h_1 = x_{1k}(z_1, z_2); \quad h_2 = x_{2k}(z_1, z_2).$$

Les composantes x_{ij}^* du programme optimal cherché X^* sont définies par les formules de récurrence

$$\left.\begin{aligned} x_{ij}^* = x_{ij}(a_1 - x_{1n}^* - \ldots - x_{1,j+1}^*, \; a_2 - x_{2n}^* - \ldots - x_{2,j+1}^*) \\ i = 1, 2; \quad j = 1, \ldots, n-1; \\ x_{in}^* = x_{in}(a_1, a_2), \quad i = 1, 2. \end{aligned}\right\} \tag{5.32}$$

Nous laissons au lecteur la justification de l'algorithme exposé, de même que l'évaluation du volume de travail nécessaire pour sa réalisation (cf. exercice 17). Dans le problème de distribution, le volume de travail exigé par la méthode de la programmation dynamique est conditionné essentiellement par le nombre m de contraintes (5.26) (dans le cas considéré $m = 2$).

Il est rationnel d'utiliser cette méthode pour $m \leqslant 3$.

Les exemples analysés illustrent le schéma général d'application de la programmation dynamique aux problèmes en nombres entiers. On voit que cette méthode se ramène à définir certaines fonctions associées par des relations de récurrence. On a intérêt à choisir les fonctions et leurs arguments en tenant compte du caractère particulier du problème, afin que les fonctions adoptées aient le moins de variables possible.

5.5. Ce chapitre donne la description des méthodes exactes les plus efficaces de programmation en nombres entiers actuellement connues. Toutefois, même ces méthodes ne permettent de résoudre que des problèmes de taille relativement petite.

Il en est ainsi parce que le nombre d'itérations auquel nous nous heurtons en essayant de résoudre un problème de programmation en nombres entiers d'une taille réelle du point de vue pratique peut s'avérer excessivement grand. Malheureusement, ceci constitue souvent un obstacle insurmontable, même pour les ordinateurs modernes. Voilà pourquoi les méthodes de programmation en nombres entiers ont besoin d'être perfectionnées et développées. L'actualité et la difficulté des problèmes qui lui sont liés font de la programmation en nombres entiers l'une des orientations les plus prometteuses et les plus intéressantes de la programmation mathématique.

A l'heure actuelle, on a fort besoin de nouveaux algorithmes de programmation en nombres entiers susceptibles d'assurer une optimisation plus rapide. L'obtention de tels algorithmes n'est possible apparemment qu'à la condition d'une étude minutieuse des particularités du problème étudié. La mise au point de méthodes de résolution de certaines classes spéciales de problèmes en nombres entiers est l'une des lignes de recherche les plus importantes et, à ce qu'il nous semble, les plus fécondes dans la programmation en nombres entiers. Il faut également accumuler l'expérience, pauvre pour l'instant, de la mise en œuvre des méthodes disponibles.

La matière de ce chapitre est en grande partie consacrée aux méthodes de programmation linéaire en nombres entiers qui ont à leur base l'idée de « troncature ». Ces derniers temps l'intérêt pour les méthodes dites combinatoires de résolution des problèmes discrets (y compris des problèmes de programmation linéaire en nombres entiers) s'est nettement renforcé. Les méthodes combinatoires sont liées à un traitement de toutes les variantes possibles du problème. Chacune de ces méthodes se caractérise par un ordre bien défini de révision des variantes et par des règles d'élimination permettant de mettre en évidence, au cours de la procédure, les variantes que l'on sait être non optimales, sans les avoir examinées au préalable. La méthode est naturellement d'autant plus efficace que le volume de travail nécessaire pour réaliser le traitement est plus petit par suite de l'application des règles d'élimination. Parmi les ouvrages consacrés aux méthodes spéciales de type combinatoire signalons [41], [47], [4]. Dans [79], [80] on trouve des algorithmes de programmation en nombres entiers associés à un traitement ordonné des variantes possibles.

Nous n'avons pas abordé ici les méthodes approchées, bien qu'on trouve dans la littérature les descriptions de quelques procédés approchés simples. La raison n'en est absolument pas que ces procédés ne présentent aucun intérêt. Bien au contraire, le grand volume de travail exigé par les méthodes exactes leur donne une énorme importance pratique. Mais notre omission est due au fait que l'élaboration et l'expérience d'application des algorithmes approchés sont encore à l'état embryonnaire.

Un certain avenir semble promis au schéma général suivant de formation d'algorithmes approchés des problèmes discrets. A chaque variante on associe un ensemble déterminé (pas très étendu) de variantes qui composent le « voisinage » de la variante en question. Le traitement direct des éléments du voisinage permet soit de passer à une meilleure variante, soit de fixer l'optimalité locale de la variante considérée. Dans le premier cas, on poursuit la recherche de la meilleure variante. Dans le deuxième, on fait un choix aléatoire de la variante qu'il convient de prendre pour le départ d'un nouveau processus d'amélioration locale. Ainsi, chacune des méthodes qui rentrent dans ce schéma est définie par sa notion de voisinage et par son mécanisme de recherche aléatoire. Lors de l'élaboration d'une méthode, il est naturellement souhaitable d'indiquer également une règle définissant la fin de la procédure et qui soit fondée sur une évaluation statistique de l'écart du meilleur des optimums locaux par rapport à la variante optimale recherchée. Un des algorithmes de ce type, appliqué à un problème particulier (le problème de distribution), est proposé dans [24].

Parmi les travaux contenant une description des méthodes de programmation en nombres entiers signalons également [46], [53], [71], [72], [73].

Exercices

1. Montrer l'équivalence des conditions (1.27) et (1.28)-(1.30).

2. Ramener le problème général de programmation en nombres entiers aux variables bornées à un problème aux conditions d'intégralité du type

$$x_j = 0 \quad \text{ou} \quad 1.$$

3. Utiliser les recommandations du paragraphe 1.4 pour ramener, avec un degré de précision donné d'avance, le problème de maximisation d'une fonction du type

$$\sum_{j=1}^{n} \varphi_j(x_j),$$

aux arguments bornés et associés par un système quelconque de conditions linéaires, à un problème de programmation linéaire (totalement) en nombres entiers.

4. Montrer que si $\varphi_j(x_j)$ sont des fonctions convexes vers le haut, la condition supplémentaire d'intégralité du problème équivalent de l'exercice 3 est superflue.

5. Décrire à l'aide d'un système de contraintes linéaires et de conditions d'intégralité le polyèdre M qui est une réunion des polyèdres convexes M_i, $i = 1, 2, \ldots, m$. Les systèmes de contraintes des polyèdres M_i sont supposés connus.

6. Démontrer le théorème 2.3 en se servant des théorèmes 2.1 et 2.2.

7. Prouver que l'algorithme du § 3 est fini en utilisant la règle de choix de la ligne l énoncée au paragraphe 3.6.

8. Résoudre à l'aide de l'algorithme du § 3 le problème en nombres entiers qui consiste à maximiser la forme linéaire

$$5x_1 + 2x_2 - 3x_3 + 2x_4 + 3x_5 - x_6$$

sous les conditions

$$\begin{aligned} 5x_1 + 6x_2 + 4x_3 + 2x_4 - 3x_5 + 5x_6 &= 11, \\ 5x_1 + 5x_2 + 7x_3 \qquad\quad + 3x_5 + 5x_6 &= 10, \\ 2x_1 + 2x_2 + 2x_3 \qquad\quad + 3x_5 \qquad\quad &= 4; \end{aligned}$$

$x_j \geqslant 0$ entier, $j = 1, 2, \ldots, 5$.

9. Démontrer que l'algorithme de programmation partiellement en nombres entiers du § 4 est fini sous les conditions du théorème 3.1 et sous l'hypothèse que la forme linéaire du problème est définie seulement par les variables entières (utiliser les raisonnements des paragraphes 3.4, 3.5).

10. Utiliser l'algorithme du § 4 pour résoudre le problème qui diffère du problème de l'exercice 8 par x_4 supposé non entier.

11. Résoudre le problème de l'exercice 8 en appliquant la règle de formation des contraintes additionnelles exposée au paragraphe 4.4.

12. Montrer que les relations (5.6)-(5.8) conduisent à la valeur minimale de α telle que l'élément directeur de la transformation à ligne directrice (5.5) soit égal à -1.

13. Résoudre par la méthode du paragraphe 5.1 le problème de maximisation de la forme linéaire

$$3x_1 - 3x_2 - 3x_3 - 4x_4 + x_5 - 2x_6 + x_7 + x_8$$

sous les conditions

$$\begin{aligned} 2x_1 - x_2 - 4x_3 - x_4 + 3x_5 - 5x_6 + x_7 &= -1, \\ 5x_1 - x_2 + 6x_3 - 2x_4 - x_5 + 4x_6 + x_8 &= 2, \\ x_j \geqslant 0 \text{ entier}, \quad j = 1, 2, \ldots, 8, \end{aligned}$$

en adoptant A_7, A_8 pour base de départ.

14. Donner un exemple de problème en nombres entiers ayant sa solution en un sommet du polyèdre des conditions et qui, néanmoins, ne peut être résolu par la méthode considérée au paragraphe 5.3 (appliquer les conditions du théorème 5.2).

15. Justifier la relation de récurrence (5.23).

16. Résoudre par la méthode de la programmation dynamique le problème de maximisation de la forme linéaire

$$2x_1 + x_2 + 4x_3 + 3x_4 + 10x_5 + 3x_6 + x_7 + 11x_8 + x_9 + 3x_{10}$$

sous les conditions

$$2x_1 + 3x_2 + 2x_3 + 4x_4 + 5x_5 + 7x_6 + x_7 + 6x_8 + 4x_9 + 2x_{10} \leqslant 13;$$

$$x_j \geqslant 0 \text{ entier}, \quad j = 1, 2, \ldots, 10.$$

17. Donner la justification de la méthode de résolution du problème de distribution en nombres entiers fondée sur les principes de la programmation dynamique et évaluer le volume de travail exigé par cette méthode.

Chapitre 6

PROGRAMMATION STOCHASTIQUE

La *programmation stochastique* est une branche de la programmation mathématique qui a pour objet les méthodes de résolution des problèmes de gestion et de planification dans des conditions de risque ou d'indétermination.

Les organes de planification sont loin d'avoir toujours à leur disposition l'information nécessaire pour la planification de l'économie ou pour la gestion de la production. Le fonctionnement des installations automatiques s'accompagne généralement d'incidents imprévus dont les lois statistiques ne peuvent pas toujours être définies et prises en compte lors du calcul des actions de commande. Ainsi, dans les modèles de programmation mathématique auxquels se ramènent les problèmes de planification et de gestion, certains paramètres (ou tous les paramètres) du facteur de qualité et des contraintes peuvent être indéterminés ou aléatoires. Dans certains cas, l'expérience, la statistique et l'analyse des processus qui déterminent la modification des données initiales permettent d'établir telles ou telles caractéristiques probabilistes des paramètres du problème. Dans d'autres cas, aucun fait ne permet de juger des principes statistiques susceptibles de faire varier les valeurs supposées des paramètres considérés. Les situations correspondant au premier cas sont appelées situations comportant un risque. Celles qui rentrent dans la deuxième catégorie sont dites indéterminées.

Malgré des recherches assez intenses dans ce domaine, la programmation stochastique est encore une branche faiblement développée de la programmation mathématique. En programmation stochastique, plus que dans les autres parties de la théorie des problèmes extrémaux conditionnels, des difficultés considérables guettent le chercheur non seulement (et, peut-être, non point 'ant) lors de l'élaboration de méthodes de résolution des problèmes, mais aussi lors de la formulation des problèmes dans lesquels il est parfois nécessaire de refléter des aspects assez délicats d'une situation, aspects qui surgissent lors de la planification et de la gestion dans des conditions de risque et d'indétermination.

La voie à première vue naturelle de l'analyse des problèmes de programmation stochastique, qui consiste à remplacer les paramètres aléatoires par leurs valeurs moyennes et calculer les programmes optimaux des problèmes déterministes ainsi obtenus, est loin de conduire toujours à une solution admissible. D'où la nécessité d'élaborer des modèles de gestion dans des conditions d'information incomplète, ainsi que des méthodes assurant leur exploration.

Le présent chapitre expose brièvement les différentes formulations de la programmation stochastique et les modes possibles de leur étude. Le § 1 est consacré à la classification des problèmes de programmation stochastique. La division des problèmes stochastiques en classes suivant divers critères permet de faire correspondre aux différents modèles de gestion dans des conditions d'information incomplète les formulations des problèmes de programmation stochastique qui sont les plus naturelles pour eux. Les §§ 2 à 4 ont pour objet l'étude de trois énoncés assez généraux de problèmes de programmation stochastique qui diffèrent entre eux par la définition de la notion de programme du problème. Dans le § 5 sont donnés les critères du programme optimal d'un problème de programmation stochastique en fonction des solutions des problèmes déterministes engendrés par ce problème. Le § 6 est consacré à la programmation dite markovienne, branche particulière mais importante pour les applications de la programmation stochastique.

§ 1. Classification des problèmes de programmation stochastique

1.1. Considérons le problème de programmation linéaire qui consiste à maximiser la forme linéaire

$$L = (C, X) \to \max \tag{1.1}$$

sous les conditions

$$AX \leqslant B, \tag{1.2}$$

$$X \geqslant 0. \tag{1.3}$$

Dans le cas général, les composantes du vecteur C de la forme linéaire, les éléments de la matrice des conditions A et les composantes du vecteur des contraintes B peuvent être des nombres aléatoires. Avec cela, les caractéristiques de la distribution des grandeurs aléatoires peuvent être connues (cas du risque) ou inconnues (cas de l'indétermination).

Examinons d'abord le problème où le polyèdre convexe des conditions (1.2)-(1.3) est déterministe et le vecteur C de la forme linéaire, aléatoire. Pour que le problème soit correctement énoncé, il est nécessaire de définir ce qu'il convient d'entendre par facteur de qualité de la résolution du problème. Souvent il semble logique de

choisir comme fonction économique du problème l'espérance mathématique de la forme linéaire (1.1). Dans de tels cas, le problème se ramène à un problème déterministe courant à forme linéaire $(\overline{C}, X)$ (où $\overline{C}$ est l'espérance du vecteur C) et aux conditions (1.2)-(1.3).

Lorsque les éléments aléatoires du problème sont constitués par la matrice des conditions ou le vecteur des contraintes, on voit apparaître des difficultés supplémentaires.

Lorsque les éléments aléatoires sont ceux de la matrice A ou du vecteur B, les relations(1.1)-(1.3) ne peuvent pas encore être considérées comme la formulation du problème. Pour que le problème soit correctement posé, il faut définir en plus ce qu'on entend par programme du problème, et dans quel sens il faut comprendre la notion du facteur de qualité de la résolution.

Les formulations des problèmes de programmation stochastique adaptées aux problèmes de programmation linéaire (1.1)-(1.3) peuvent être différentes. Elles se distinguent principalement par la définition des notions de programme et de solution. La préférence peut aller à telle ou telle formulation, suivant l'allure particulière de l'application. Il est naturel de prendre comme critères fondamentaux de la classification les définitions possibles du programme et de la solution. Comme nous le verrons plus loin, il est bon d'introduire encore d'autres critères de classification.

1.2. Commençons par classer les problèmes d'après la définition de la notion de leur programme.

En fonction du contenu concret du problème de choix du comportement dans les conditions d'indétermination ou de risque, il est avantageux de considérer comme programme une collection déterministe ou aléatoire de paramètres de commande. Dans l'un et dans l'autre cas, il y a différentes possibilités pour définir le programme.

Considérons d'abord le cas où le programme est un vecteur déterministe.

On peut indiquer des problèmes dans lesquels la violation d'une ou plusieurs de leurs contraintes, lors de la réalisation des conditions aléatoires du problème, conduit à des conséquences très fâcheuses, si ce n'est catastrophiques. Dans de tels problèmes, il est rationnel de ne comprendre par programme que les vecteurs X qui satisfont aux contraintes pour toute matrice A et tout vecteur B à probabilité positive. Une telle approche conduit à ce que l'on appelle une formulation serrée du problème de programmation stochastique.

La formulation serrée des problèmes de programmation stochastique exige l'adoption d'une solution en une seule étape et ne permet pas de corriger la solution adoptée lorsqu'apparaissent des informations supplémentaires sur la réalisation des conditions du problème.

Les conditions d'un tel problème s'écrivent

$$A(q)\,X \leqslant B(q) \quad \text{pour tout} \quad q \in Q, \quad X \geqslant 0. \tag{1.4}$$

q est ici le paramètre aléatoire ou la collection de paramètres aléatoires déterminant la réalisation des éléments aléatoires des conditions du problème ; Q, l'ensemble des valeurs de q à probabilité positive. (L'ensemble Q ici est supposé fini.)

Le choix d'une réalisation de q s'appelle généralement détermination d'un *état de la nature*, et l'ensemble Q, *ensemble des états possibles de la nature*.

La formulation serrée d'un problème de programmation stochastique nécessite une solution qui appartienne à tous les ensembles convexes des conditions à probabilité non nulle. Dans ces conditions on ne distingue pas les domaines de définition du problème à probabilité très forte ou très faible. L'intersection des polyèdres convexes, qui délimitent le domaine de définition du problème, peut s'avérer vide. La formulation serrée du problème n'a alors aucun sens.

Il existe de nombreux problèmes stochastiques dans lesquels il est plus avantageux de ne pas éliminer de l'exploration les collections des paramètres de commande associés à des variables d'écart de certaines conditions ayant lieu dans des cas de réalisations peu fréquentes de paramètres définissant les contraintes du problème. Il est beaucoup plus avantageux d'instituer une pénalité pour l'infraction aux conditions et en tenir compte dans le facteur de qualité de la solution du problème. Le montant de la pénalité doit naturellement être déterminé par la grandeur de la variable d'écart. Cette formulation du problème semble être plus souple et souvent plus réaliste.

Dans ce cas la procédure de résolution d'un problème stochastique peut être pensée comme divisée en deux stades. Le premier consiste à choisir un certain vecteur non négatif X qui ne vérifie pas nécessairement toutes les conditions du problème relatives à tous les états de la nature possibles. Ensuite, on fixe la réalisation de la matrice aléatoire A et du vecteur aléatoire B et, selon la variable d'écart $AX - B$, on introduit le vecteur ξ qui corrige la solution adoptée. Cette possibilité de corriger la solution supprime les difficultés relatives à la définition de la notion de « programme d'un problème stochastique ».

Les problèmes stochastiques ainsi énoncés s'appellent *problèmes à deux étapes* de programmation stochastique.

Il est logique d'énoncer les problèmes à deux étapes comme des problèmes de recherche du minimum. La minimisation porte sur l'espérance mathématique de la somme des valeurs de la fonction économique du problème relativement à la solution adoptée et à la pénalité définie par les variables d'écart pour non-conformité de

la modalité adoptée aux conditions du problème. Lorsque la pénalité devient infiniment grande, la formulation du problème à deux étapes devient une formulation serrée.

On considère également d'autres méthodes de définition du programme d'un problème de programmation stochastique.

Pour beaucoup de problèmes stochastiques engendrés par les applications, il apparaît naturel de remplacer les conditions (1.2) par des contraintes probabilistes du type

$$P\left(\sum_{j=1}^{n} a_{ij}x_j \leqslant b_i\right) \geqslant p_i, \quad 0 < p_i \leqslant 1, \quad i = 1, 2, \ldots, m, \tag{1.5}$$
$$X \geqslant 0.$$

Cette écriture signifie que dans la formulation adoptée on entend par programme du problème un vecteur $X \geqslant 0$ qui vérifie la i-ème inégalité du système (1,2) avec une probabilité P au moins égale à la quantité donnée p_i. Lorsque la violation des contraintes d'un problème à deux étapes conduit à une pénalité importante, il est naturel de choisir des valeurs de p_i voisines de 1. Quant aux contraintes moins essentielles, qui donnent lieu à des variables d'écart peu coûteuses, il est rationnel de les associer à des valeurs des p_i plus petites. Dans la formulation serrée des problèmes de programmation stochastique, toutes les composantes p_i sont égales à l'unité, indépendamment de l'importance de telle ou telle contrainte. Ainsi, la formulation serrée peut être définie dans le cas général indépendamment de l'allure finie ou infinie de l'ensemble Q des états de la nature.

Appelons dans ce qui suit *problèmes à contraintes probabilistes* les problèmes de programmation stochastique aux conditions du type (1.5).

Dans les formulations examinées jusqu'à présent, on entendait par programme et par solution d'un problème les collections déterministes de paramètres de commande. On peut indiquer des applications dans lesquelles on a intérêt à associer la solution du problème aux réalisations concrètes de ses conditions aléatoires. Le programme et la solution sont alors des collections aléatoires de paramètres de commande dont les lois statistiques sont définies par les conditions du problème.

Il arrive souvent que le contenu concret du problème suggère le type des liaisons supplémentaires naturelles qui expriment chaque programme (y compris le programme optimal) à l'aide de la réalisation des conditions aléatoires (éléments de la matrice A et composantes des vecteurs B et C) et la collection D des paramètres déterministes :

$$X = \Phi(A, B, C, D).$$

Une telle formulation du problème impose le calcul de la collection D des paramètres de commande déterministes.

Un système dans lequel la commande est régie de la façon indiquée est mieux adapté à la variation des conditions du problème. Il n'existe pas pour la commande d'un tel système de règle unique, commune à tous les sommets de l'ensemble des états de la nature possibles; cette commande change avec chaque réalisation des conditions du problème, en intervenant avec souplesse dans la modification des états de la nature.

Dans les problèmes de programmation stochastique où seul le vecteur des contraintes est aléatoire, il s'avère souvent utile d'associer le programme X à la réalisation aléatoire du vecteur B par une relation linéaire de la forme

$$X = DB, \tag{1.6}$$

où D est la matrice cherchée, de dimensions $n \times m$.

Il existe des situations dans lesquelles il n'y a aucune raison de donner d'avance le type de la relation fonctionnelle (règle de résolution) qui associe la solution aux réalisations des paramètres des conditions du problème. D'où l'intérêt que présentent les formulations de programmation stochastique dans lesquelles la règle de résolution n'est pas donnée mais s'obtient au terme de la résolution du problème.

Remarquons que le passage de la recherche de la solution sous la forme d'un vecteur déterministe à la recherche de la règle de résolution étend le domaine de définition du problème et aussi, dans le cas général, la marge de variation du facteur de qualité de la solution.

Dans ces conditions, le maximum accessible de la fonction économique ne peut qu'augmenter et le minimum accessible ne peut que diminuer. C'est pourquoi, dans tous les cas où les règles de résolution ont un sens et peuvent être interprétées en termes cohérents du problème considéré, il est rationnel de définir la solution d'un problème de programmation stochastique sous la forme d'un vecteur aléatoire.

Le calcul du vecteur aléatoire X se ramène ainsi au calcul de la matrice déterministe D.

Dans ce qui suit, nous rencontrerons de nouveau des énoncés de problèmes dans lesquels il est naturel de prendre le vecteur aléatoire X comme solution du problème.

1.3. Dans les formulations des problèmes stochastiques on doit spécifier non seulement ce qu'on entend par programme, mais encore ce qu'il faut considérer comme solution du problème. A cet égard présente un intérêt la classification des problèmes de programmation stochastique d'après le facteur de qualité de la résolution du problème.

Voici les facteurs qui se présentent tout naturellement:

a) espérance mathématique de la valeur de la forme linéaire;

b) variance de la forme linéaire;

c) combinaison linéaire de l'espérance mathématique et de la variance de la forme linéaire;

d) probabilité d'une forme linéaire dépassant un certain seuil fixé;

e) espérance mathématique de la fonction d'utilité de la forme linéaire;

f) maximin de la forme linéaire (le maximum est pris sur l'ensemble des programmes X et le minimum sur les états de la nature q admissibles);

g) maximin de l'espérance mathématique de la forme linéaire (le maximum est pris sur les distributions F_X des programmes X définis sur l'ensemble des programmes du problème, et le minimum sur les distributions $F_{A.B.C}$ des éléments des conditions du problème ou, ce qui revient au même, sur les distributions F_q des états de la nature données sur l'ensemble Q des états de la nature admissibles).

En choisissant la fonction économique comme critère pour classer les problèmes de programmation stochastique, il est naturel de considérer sept classes, conformément aux facteurs de qualité de la solution qui ont été dénombrés.

Les problèmes dans lesquels le facteur de qualité est l'espérance mathématique de la valeur de la forme linéaire s'appellent E-modèles.

Les problèmes qui consistent à minimiser la variance de la forme linéaire portent le nom de V-modèles. Appelons EV-modèles les problèmes qui imposent l'optimisation de l'espérance mathématique de la forme linéaire compte tenu de la pénalité due à la variance. Le cas de l'optimisation de la probabilité pour la forme linéaire de dépasser un certain seuil est appelé P-modèle.

Le nom de U-modèles est attribué aux problèmes dont le facteur de qualité de la solution est la fonction d'utilité de la forme linéaire.

Nous appelons Mm-modèle les problèmes dans lesquels le maximin de la forme linéaire est une fonction économique, et MmE-modèle les problèmes dans lesquels le facteur de qualité de la solution est le maximin de l'espérance mathématique de la forme linéaire.

Constatons que pour un Mm-modèle la solution du problème est un vecteur déterministe. Pour les MmE-modèles, la solution du problème est généralement un vecteur aléatoire dont les caractéristiques statistiques sont déterminées d'après les caractéristiques probabilistes des conditions du problème. Le choix, en qualite de fonction économique de la solution, du maximin de l'espérance mathématique de la valeur de la forme linéaire ramène le problème de programmation stochastique à un problème de jeu. L'écriture de la formulation en termes de jeu des problèmes de programmation

stochastique devient plus naturelle si l'on considère comme fonction de payement non la valeur de la forme linéaire mais la différence entre cette valeur et la pénalité pour la variable d'écart du programme.

1.4. Considérons un peu plus en détail, afin de ne pas y revenir par la suite, la formulation en termes de jeu du problème de programmation stochastique [43].

Le problème (1.1)-(1.3), où les éléments de la matrice des conditions A, du vecteur des contraintes B et du vecteur de la forme linéaire C sont aléatoires, est considéré ici comme un jeu à deux personnes à somme nulle. Le premier joueur est celui qui prend la décision. Il s'efforce d'obtenir le plus grand gain moyen possible. Il s'oppose à la nature dont les états interviennent pour réduire au minimum le gain moyen du premier joueur. Pour chaque état de la nature q et pour chaque choix de X, on entend par gain la différence entre la valeur correspondante de la forme linéaire et la pénalité pour la variable d'écart du programme:

$$g[X, A(q), B(q), C(q)] = \\ = \sum_{j=1}^{n} c_j(q)\, x_j - \sum_{j=1}^{m} \varphi_i \left[\sum_{j=1}^{n} a_{ij}(q)\, x_j - b_i(q) \right].$$

Ici $\varphi_i(y)$ $(i = 1, 2, \ldots, m)$ est une fonction continue non décroissante de y, égale à zéro si $y \leqslant 0$.

L'ensemble Q des états de la nature possibles q et le domaine M des programmes admissibles X sont supposés donnés d'avance. L'ensemble Q définit la partie de l'espace euclidien de dimensions $mn + m + n$ qui correspond au domaine admissible de variation des éléments a_{ij}, b_i, c_j des conditions du problème. Appelons ce domaine ensemble N.

Soient S l'ensemble des distributions F_X du vecteur X définies sur M et T l'ensemble des distributions communes $F_{A,B,C}$ de la matrice A et des vecteurs B et C définies sur N. Il a été démontré que pour des hypothèses assez peu restrictives relatives aux ensembles M et N (hypothèses qui se ramènent à supposer ces ensembles compacts), il existe des $F_X \in S$ et des $F_{A,B,C} \in T$ sur lesquels on obtient la solution du jeu

$$\max_{F_X \in S;\ F_{A,B,C} \in T} \min \int_{M \times N} g(\xi, \alpha, \beta, \gamma)\, dF_X(\xi)\, dF_{A,B,C}(\alpha, \beta, \gamma) =$$

$$= \min_{F_{A,B,C} \in T;\ F_X \in S} \max \int_{M \times N} g(\xi, \alpha, \beta, \gamma)\, dF_X(\xi)\, dF_{A,B,C}(\alpha, \beta, \gamma) =$$

$$= \int_{M \times N} g(\xi, \alpha, \beta, \gamma)\, dF_X^*(\xi)\, dF_{A,B,C}^*(\alpha, \beta, \gamma).$$

F_X^* et $F_{A,B,C}^*$ sont ici les stratégies optimales de celui qui prend la décision et de la nature.

Si le problème (1.1)-(1.3) est résolu dans des conditions d'indétermination, quand la fonction $F_{A,B,C}$ n'est pas connue d'avance, T est un ensemble de toutes les distributions possibles définies sur N, et la solution F_{X*} du problème stochastique détermine en général une stratégie mixte.

Si le problème de programmation linéaire est résolu dans des conditions de risque, quand la fonction $F_{A,B,C}$ est connue d'avance, l'ensemble T se réduit à ce seul élément. La stratégie optimale du premier joueur se trouve être alors une stratégie pure. On peut choisir comme ensemble M, par exemple, l'ensemble des vecteurs $X \geqslant 0$ qui satisfont à la condition (1.5). Pour $p_i = 1$, la pénalité disparaît de l'expression du gain g. Nous aboutissons à la formulation serrée du problème de programmation stochastique,

$$\int_M (\bar{C}, \xi)\, dF_X(\xi) = (\bar{C}, X) \to \max,$$

sous la condition $X \in M$.

On voit que les différentes formulations d'un problème de programmation stochastique sont liées surtout aux diverses définitions du programme et de la solution du problème. Il existe d'autres facteurs encore qui interviennent dans l'approche à la formulation et à l'exploration du problème. La question de savoir quelles sont parmi les formulations des problèmes stochastiques celles qu'il convient de reconnaître comme meilleures ou comme moins bonnes n'a de sens qu'appliquée à des cas concrets. Dans des situations où chaque erreur de calcul peut être fatale, il est naturel de recourir à la formulation serrée du problème. Si dans ces conditions la formulation serrée n'a plus de sens, il est rationnel de passer à un problème à contraintes probabilistes du type (1.5), aux valeurs des p_i aussi proches que possible de l'unité. Dans les situations où une extension du domaine de définition du programme justifie l'espoir d'un gain fourni par la forme linéaire, ce gain dépassant la pénalité due à la variable d'écart, il est naturel d'opter pour une formulation à deux étapes (problème de décision). On peut enfin indiquer des situations dans lesquelles l'effet de la commande dans les conditions d'indétermination ne se manifeste que pour de multiples répétitions d'une certaine procédure. Dans les cas où les décisions sont prises chaque fois dans des conditions identiques, la formulation en termes de jeu des problèmes de programmation stochastique est justifiée et peut être recommandée.

1.5. Les problèmes de programmation stochastique peuvent être classés également en problèmes statiques et dynamiques. Dans les problèmes statiques, le caractère stochastique des conditions

est déterminé par des grandeurs aléatoires, dans les problèmes dynamiques, par des fonctions aléatoires ou des suites aléatoires. Les lois de variation des caractéristiques statistiques des fonctions ou des suites aléatoires doivent être prises en compte aussi bien dans la formulation du problème que dans la construction des méthodes d'analyse. L'optimisation des modèles probabilistes dynamiques engendrés par les processus ou chaînes de Markov s'appelle *programmation markovienne*.

Ces dernières années on a vu apparaître également des ouvrages consacrés à des problèmes stochastiques à étapes multiples aux contraintes probabilistes conditionnelles et inconditionnelles. Les paramètres aléatoires des conditions de ces problèmes, paramètres intervenant à des stades différents, ne constituent pas nécessairement une chaîne de Markov.

Pour certaines classes de problèmes stochastiques à étapes multiples, on a construit et exploré des règles de résolution optimales.

On peut indiquer également d'autres critères de classification des problèmes de programmation stochastique, différents de ceux mentionnés ci-dessus. Nous ne les donnons pas, parce que l'expérience acquise dans la formulation et la résolution de ces problèmes étant manifestement insuffisante, elle ne fournit pas encore de raison bien fondée pour choisir une classification qui reflète les voies de développement de cette discipline.

§ 2. Formulation serrée des problèmes de programmation stochastiques

2.1. Pour fixer les idées, considérons les problèmes à une étape du type E-modèle. Désignons comme auparavant les états de la nature aléatoires par q et l'ensemble des états de la nature par Q. Le problème à une étape consiste à choisir le vecteur X sur lequel est atteint le maximum de l'espérance mathématique de la valeur de la forme linéaire

$$E\,(C\,(q),\ X) \tag{2.1}$$

avec

$$A\,(q)\,X \leqslant B\,(q), \quad q \in Q, \tag{2.2}$$

$$X \geqslant 0. \tag{2.3}$$

Soit l'ensemble fini Q des états de la nature.

Nous dirons que le couple $[A,\ B]$ est admissible si sa probabilité est positive. Le vecteur X, programme du problème stochastique quel que soit l'état de la nature $q \in Q$ (c'est-à-dire quel que soit le couple admissible $[A,\ B]$), est dit *programme permanent* du problème. L'ensemble S des programmes permanents d'un problème stochastique se compose ainsi des vecteurs $X \geqslant 0$ tels que leur

probabilité

$$P\,[A\,(q)\,X \leqslant B\,(q)] = 1. \tag{2.4}$$

Supposons que la matrice A° et le vecteur B° forment un couple admissible, et que le vecteur X° soit solution du problème déterministe de programmation linéaire dans lequel on demande de maximiser la forme linéaire

$$L = (\overline{C},\ X) \tag{2.5}$$

avec

$$A^\circ X \leqslant B^\circ, \tag{2.6}$$

$$X \geqslant 0, \tag{2.7}$$

où $\overline{C} = EC\,(q)$ est l'espérance mathématique du vecteur des coefficients de la forme linéaire.

Il est clair que si X° est un programme permanent du problème stochastique, il en sera aussi sa solution. Ainsi, l'une des approches à la résolution d'un problème à une étape consiste à rechercher un programme permanent X° qui serait solution du problème déterministe de programmation linéaire pour un certain couple admissible $[A^\circ, B^\circ]$.

Si $[A^\circ, B^\circ]$ n'est pas un couple admissible, alors de ce que le programme optimal X du problème (2.5)-(2.7) est permanent il ne suit encore pas que X est solution du problème stochastique.

Le couple $[A^\circ, B^\circ]$ est constitué généralement de la matrice $\overline{A} = EA$ et du vecteur $\overline{B} = EB$ dont les éléments sont les espérances mathématiques des éléments correspondants $A\,(q)$ et $B\,(q)$. De plus, on suppose que le couple $[\overline{A}, \overline{B}]$ est admissible.

Soit $\overline{X} = (\overline{x}_1, \ldots, \overline{x}_n)$ le programme optimal du problème déterministe (2.5)-(2.7). Pour établir si $\overline{X}$ est solution du problème stochastique correspondant, il suffit de vérifier s'il est permanent. Lorsque tout élément de la matrice $A\,(q)$ est borné supérieurement et les composantes des vecteurs $B\,(q)$ le sont inférieurement, cette vérification ne présente pas de difficultés.

Il est facile de voir directement que le vecteur $\overline{X}$ est programme permanent du problème à une étape si

$$\max_i \left[\sum_{j=1}^{n} a_{ij}^{+}\overline{x}_j - b_i^{-}\right] \leqslant 0, \tag{2.8}$$

où a_{ij}^{+} est la borne supérieure des éléments $a_{ij}\,(q)$ et b_i^{-} la borne inférieure des $b_i\,(q)$ pour tout $q \in Q$.

Cette méthode de résolution des problèmes à une étape qui débute par le calcul du programme optimal du problème déterministe

$$\overline{C}X \to \max$$

sous les conditions

$$\bar{A}X \leqslant \bar{B},$$
$$X \geqslant 0,$$

s'appelle *résolution par rapport aux moyennes.*

2.2. Il n'y a bien entendu aucune garantie pour que, partant d'un couple admissible $[A^0, B^0]$, et, notamment, de $[\bar{A}, \bar{B}]$, nous obtenions forcément un programme permanent du problème stochastique. Il peut y avoir des cas où la préférence doit aller à la méthode suivante de résolution du problème, qui est plus délicate.

Pour chaque état de la nature fixe q, les conditions (2.2)-(2.3) découpent dans l'espace des variables $\bar{X}$ un polyèdre convexe. L'intersection de tous les ensembles associés aux différents états de la nature, elle aussi, est convexe. Si l'ensemble des couples $[A, B]$ est fini, la résolution du problème stochastique se ramène au calcul du programme optimal du problème déterministe suivant:

$$E\,(C,\ X) \to \max, \tag{2.9}$$

$$A_q X \leqslant B_q, \quad q = 1,\ 2,\ \ldots,\ k, \tag{2.10}$$

$$X \geqslant 0, \tag{2.11}$$

où $[A_q, B_q]$ sont les couples admissibles numérotés des matrices des conditions et des vecteurs des contraintes.

Une voie approchée de résolution des problèmes stochastiques à nombre infini d'états de la nature consiste à approximer l'ensemble des couples $[A, B]$ par un ensemble fini et à résoudre un problème du type (2.9)-(2.11).

§ 3. Problèmes à contraintes probabilistes

3.1. Dans plusieurs applications, l'écriture naturelle des contraintes d'un problème stochastique est de la forme

$$P\left(\sum_{j=1}^{n} a_{ij},\ x_j \leqslant b_i\right) \geqslant p_i,$$
$$0 < p_i \leqslant 1, \quad i = 1,\ 2,\ \ldots,\ m. \tag{3.1}$$

Le sens de cette notation a été explicité au paragraphe 1.2.

Les problèmes à contraintes probabilistes sont examinés dans une série d'articles de Charnes et Cooper. L'attitude des auteurs devant la résolution de ces problèmes est le mieux exposée dans [77].

Pour une matrice fixe A et un vecteur de contraintes aléatoire B (ce sont les seuls problèmes traités dans ce paragraphe), la condition (3.1) peut être ramenée à une forme usuelle et mise sous la forme (2.6).

Soit $\varphi\,(b_i, \ldots, b_m)$ la densité multidimensionnelle de distribution des composantes b_i du vecteur de contraintes aléatoire B.

La densité de distribution de la composante b_i s'écrit

$$\varphi_i(b_i) = \int_{-\infty}^{\infty} \overset{(m-1)}{\cdots\cdots} \int_{\infty}^{\infty} \varphi(b_1, \ldots, b_m) \prod_{j \neq 1} db_j.$$

Calculons $\widetilde{b}_i$ d'après l'équation

$$\int_{\widetilde{b}_i}^{\infty} \varphi_i(b_i)\, db_i = p_i. \tag{3.2}$$

Il est clair que la relation

$$P\left(\sum_{j=1}^{n} a_{ij}x_j \leqslant b_i\right) \geqslant p_i \tag{3.3}$$

est équivalente à l'inégalité

$$\sum_{j=1}^{n} a_{ij}x_j \leqslant \widetilde{b}_i, \tag{3.4}$$

où $\widetilde{b}_i$ est calculé d'après (3.2).

L'équivalence des relations (3.3) et (3.4) permet de ramener le problème stochastique de programmation linéaire du type

$$E(C, X) \to \max,$$
$$P(AX \leqslant B) \geqslant p,$$

où les vecteurs B et C sont aléatoires et où la matrice A est déterministe, au problème déterministe de programmation linéaire

$$(\overline{C}, X) \to \max,$$
$$AX \leqslant \widetilde{B}.$$

3.2. Dans les problèmes de programmation stochastique il faut souvent (cf. paragraphe 1.2) que la solution X vérifie, en plus des conditions du type (3.3) ou (3.4), les relations du type

$$X = \Phi(A, B, C),$$

où Φ est une fonction de classe déterminée.

Dans [77], on considère comme condition supplémentaire les règles de commande du type

$$X = DB,$$

où D est une matrice déterministe inconnue de dimension $(n \times m)$ (la matrice A des conditions du problème a, comme d'habitude, la dimension $m \times n$).

Avec la règle de commande supplémentaire ainsi introduite, le programme X et la solution X^* sont des vecteurs aléatoires dont

les réalisations sont définies par les réalisations du vecteur aléatoire des contraintes.

Considérons le problème suivant de programmation stochastique. Calculer le vecteur X tel que

$$E(C, X) \to \max, \tag{3.5}$$

$$P(AX \leqslant B) \geqslant p, \tag{3.6}$$

$$X = DB. \tag{3.7}$$

(Les conditions $X \geqslant 0$ sont supposées incluses dans le système d'inégalités (3.6) avec $p_i = 1$.)

Dans le problème (3.5)-(3.7), la matrice A est déterministe alors que les vecteurs B et C sont des vecteurs aléatoires indépendants. Les composantes de chacun de ces vecteurs peuvent être corrélées entre elles.

Trouver la solution de ce problème signifie de calculer les éléments d_{ij} de la matrice D.

Transformons l'écriture du problème (3.5)-(3.7). Portons (3.7) dans le facteur de qualité de la solution du problème. L'indépendance statistique des vecteurs C et B entraîne

$$E(C, X) = \overline{(C, X)} = (\overline{C}, D\overline{B}) = \sum_{i=1}^{m} \sum_{j=1}^{n} d_{ji}\bar{b}_i\bar{c}_j,$$

où $\overline{C}$ et $\overline{B}$ sont les espérances mathématiques des vecteurs C et B.

Le facteur de qualité (3.5) est ramené de cette manière à une forme entièrement déterministe. Le problème (3.5)-(3.7) consiste à calculer la matrice D qui maximise la fonction économique

$$\overline{(\overline{C}, DB)} \tag{3.8}$$

sous les conditions

$$P(ADB \leqslant B) \geqslant p. \tag{3.9}$$

Ramenons les conditions (3.9) au type déterministe équivalent.

Désignons par $a_i = (a_{i1}, \ldots, a_{in})$ le i-ème vecteur-ligne de la matrice A et introduisons la variable aléatoire ζ_i

$$\zeta_i = \frac{(b_i - \bar{b}_i) - a_i D(B - \overline{B})}{\sqrt{E[(b_i - \bar{b}_i) - a_i D(B - \overline{B})]^2}}.$$

Les raisonnements qui suivent se font dans l'hypothèse que la variable ζ_i ou, ce qui revient ici au même, le vecteur des contraintes B sont distribués suivant une loi normale.

On voit aisément que

$$\bar{\zeta}_i = E\zeta_i = 0 \text{ et } \sigma^2_{\zeta_i} = \overline{\zeta_i^2} = 1,$$

c'est-à-dire que les grandeurs aléatoires ζ_i ont des variances moyennes nulles et unitaires.

Ecrivons la condition (3.9) séparément pour chaque contrainte:

$$P\,(a_i DB \leqslant b_i) \geqslant p_i, \quad i = 1, 2, \ldots, m. \tag{3.10}$$

Il est aisé de s'assurer par calcul direct que l'inégalité

$$a_i DB \leqslant b_i$$

est équivalente à la relation

$$\zeta_i \geqslant \zeta_i^\circ = \frac{-\bar{b}_i + a_i D\bar{B}}{\sqrt{E\,[(b_i - \bar{b}_i) - a_i D\,(B - \bar{B})]^2}}. \tag{3.11}$$

C'est pourquoi les conditions (3.10) peuvent être remplacées par des inégalités du type

$$P\,(\zeta_i \geqslant \zeta_i^\circ) \geqslant p_i, \quad i = 1, 2, \ldots, m \tag{3.12}$$

Par hypothèse, les ζ_i sont distribués d'après une loi normale. Par suite, il résulte de (3.12) que

$$\frac{1}{\sqrt{2\pi}} \int\limits_{\zeta_i^\circ}^{\infty} e^{-\frac{t^2}{2}}\, dt \geqslant p_i,$$

ou

$$1 - \Phi\,(\zeta_i^\circ) \geqslant p_i,$$

où $\Phi(\zeta)$ est la fonction de Laplace.

Cette dernière inégalité est équivalente à la relation

$$\zeta_i^\circ \leqslant \Phi^{-1}\,(1 - p_i) = -k_i. \tag{3.13}$$

Avec $p_i > \frac{1}{2}$ (et c'est le seul cas qui présente quelque intérêt dans les problèmes pratiques),

$$k_i > 0.$$

Sous les hypothèses adoptées, les nombres k_i dépendent seulement des probabilités données p_i et ne dépendent pas des éléments de la matrice recherchée D.

Portons l'expression de ζ_i° tirée de (3.11) dans l'inégalité (3.13). Nous obtiendrons l'écriture suivante des contraintes du problème initial:

$$\bar{b}_i - a_i D\bar{B} \geqslant k_i \sqrt{E\,[(b_i - \bar{b}_i) - a_i D\,(B - \bar{B})]^2}.$$

Introduisons des variables additionnelles z_i telles que

$$\bar{b}_i - a_i D\bar{B} \geqslant z_i \geqslant k_i \sqrt{E\,[(b_i - \bar{b}_i) - a_i D\,(B - \bar{B})]^2}.$$

Cette dernière inégalité est équivalente au système d'inégalités

$$a_i D\bar{B} + z_i \leqslant \bar{b}_i, \tag{3.14}$$

$$-k_i^2 E\,[(b_i - \bar{b}_i) - a_i D\,(B - \bar{B})]^2 + z_i^2 \geqslant 0, \tag{3.15}$$

$$z_i \geqslant 0, \tag{3.16}$$

$$i = 1, 2, \ldots, m.$$

Chacune des conditions du type (3.14) définit un demi-espace dans l'espace des variables d_{ij} ($i = 1, 2, \ldots, m$; $j = 1, 2, \ldots, n$) et z_i. L'intersection de ces demi-espaces pour $i = 1, \ldots, m$ correspond à un polyèdre convexe. On peut montrer (cf. exercice 5) que pour tout i le couple de conditions du type (3.15) et (3.16) découpe dans l'espace des variables d_{ij} ($i = 1, 2, \ldots, m$; $j = 1, 2, \ldots, n$) et z_i un ensemble convexe. On vérifiera aisément que cet ensemble est la nappe « supérieure » (au sens de l'axe des z_i) d'un hyperboloïde à deux nappes. L'intersection des polyèdres convexes associés aux conditions (3.14) et des nappes « supérieures » des hyperboloïdes découpés par les conditions (3.15)-(3.16) définit dans l'espace des variables d_{ij}, z_i ($i = 1, 2, \ldots, m$; $j = 1, 2, \ldots, n$) un ensemble convexe.

Le problème de programmation stochastique (3.5)-(3.7) à contraintes probabilistes se ramène ainsi à un problème déterministe non linéaire de programmation mathématique à facteur de qualité linéaire (3.8) et aux contraintes non linéaires (3.14)-(3.16) qui définissent sur l'espace des variables d_{ij}, z_i un ensemble convexe.

Un problème qui consiste à maximiser une fonction convexe vers le haut (ou minimiser une fonction convexe vers le bas) sur un ensemble convexe s'appelle problème de *programmation convexe*. Le problème (3.8), (3.14)-(3.16) constitue un problème de programmation convexe.

Récrivons le problème de programmation convexe obtenu sous une forme plus condensée. Introduisons les notations

$$\mu_i\,(D) = E\,(b_i - a_i DB), \tag{3.17}$$

$$\sigma_i^2\,(D) = E\,(b_i - a_i DB)^2, \tag{3.18}$$

$$i = 1, 2, \ldots, m.$$

Dans ces notations, le problème de programmation convexe, équivalent déterministe du problème de programmation stochastique (3.5)-(3.7), s'écrit

$$\bar{C}\bar{D}B \to \max \tag{3.19}$$

sous les conditions

$$\mu_i (D) - z_i \geqslant 0, \tag{3.20}$$

$$k_i^2 [\mu_i^2 (D) - \sigma_i^2 (D)] + z_i^2 \geqslant 0, \tag{3.21}$$

$$z_i \geqslant 0, \tag{3.22}$$

$$i = 1, 2, \ldots, m.$$

3.3. Dans la terminologie du § 1, le problème à contraintes probabilistes examiné au paragraphe précédent est un E-modèle. Si l'on choisit comme fonction économique l'espérance mathématique de l'écart-type de la forme linéaire (C, X) par rapport au nombre donné (C^o, X^o), on peut, en reprenant les mêmes raisonnements qu'au paragraphe précédent, démontrer que le problème de programmation stochastique (V-modèle)

$$E [(C, X) - (C^o, X^o)]^2 \to \min$$

sous les conditions

$$P (AX \leqslant B) \geqslant p,$$

$$X = DB$$

est équivalent au problème déterministe de programmation convexe

$$V (D) = E [CDB - (C^o, X^o)]^2 \to \min \tag{3.23}$$

sous les conditions (3.20)-(3.22). Ici, de même qu'auparavant, la matrice des conditions A est supposée fixe et les vecteurs B et C, aléatoires.

On peut également réduire à un problème de programmation convexe le problème suivant de programmation stochastique, cas particulier du P-modèle.

Soit à calculer le vecteur X maximisant la probabilité P pour que la forme linéaire (C, X) dépasse le seuil imposé (C^o, X^o),

$$P [(C, X) \geqslant (C^o, X^o)] \to \max, \tag{3.24}$$

sous la condition

$$P (AX \leqslant B) \geqslant p, \tag{3.25}$$

et la restriction supplémentaire que l'on obtienne la solution sous la forme

$$X = DB. \tag{3.26}$$

On adopte relativement à A, B et C les mêmes hypothèses qu'au paragraphe précédent.

Introduisons la variable aléatoire

$$\eta = \frac{CDB - \bar{C}D\bar{B}}{\sqrt{E [CDB - \bar{C}D\bar{B}]^2}}. \tag{3.27}$$

Nous ne considérons que les problèmes dans lesquels la distribution de η est normale. On voit aisément que

$$\overline{\eta} = 0, \quad \sigma_\eta^2 = \overline{\eta^2} = 1,$$

et que, par conséquent, la fonction de distribution de η ne dépend pas de la matrice cherchée D.

Désignons par η_0 la constante

$$\eta_0 = \frac{(C^\circ, X^\circ) - \overline{C}D\overline{B}}{\sqrt{E[CDB - \overline{C}D\overline{B}]^2}}. \tag{3.28}$$

Il est facile de s'assurer, par des calculs immédiats, qu'avec $X = DB$ la restriction (3.24) peut être mise sous la forme équivalente suivante:

$$P(\eta \geqslant \eta_0) \to \max,$$

ou en vertu de l'hypothèse adoptée sur la distribution normale de η:

$$\int_{\eta_0}^{\infty} e^{-\frac{t^2}{2}} dt \to \max.$$

Cette dernière restriction équivaut à

$$\eta_0 \to \min,$$

ou, ce qui revient au même, à

$$\frac{\overline{C}D\overline{B} - (C^\circ, X^\circ)}{\sqrt{E[CDB - \overline{C}D\overline{B}]^2}} \to \max.$$

Introduisons de nouvelles variables artificielles z_0 et W_0:

$$W_0^2 \geqslant V(D) = E(CDB - CD\overline{B})^2, \tag{3.29}$$

$$z_0 \leqslant \overline{C}D\overline{B} - (C^\circ, X^\circ). \tag{3.30}$$

Les conditions (3.25)-(3.26) du problème se transforment, comme dans le cas du E-modèle, en contraintes (3.20)-(3.22). C'est pourquoi le problème de programmation stochastique (3.24)-(3.26) se ramène au problème non linéaire déterministe suivant:

$$\frac{z_0}{W_0} \to \max \tag{3.31}$$

sous les conditions (3.29)-(3.30) et (3.20)-(3.22).

Seuls présentent un intérêt pratique les problèmes pour lesquels $V(D) > 0$. Pour ces problèmes, $W_0 > 0$ en raison de (3.29). Afin de se libérer de l'exposant homographique, introduisons la variable

$$t = \frac{1}{W_0}. \tag{3.32}$$

Adoptons les notations

$$\widetilde{D} = tD, \quad \widetilde{z}_i = tz_i, \quad \widetilde{W}_0 = tW_0 = 1,$$
$$\widetilde{V}(\widetilde{D}) = E\,[C\widetilde{D}B - t\,(C^\circ,\ X^\circ)]^2,$$
$$\widetilde{\sigma}_i^2(\widetilde{D}) = E\,[tb_i - a_i\widetilde{D}B]^2,$$
$$\widetilde{\mu}_i(\widetilde{D}) = E\,(tb_i - a_i\widetilde{D}B).$$

Avec les nouvelles variables ($\widetilde{d}_{ij} = td_{ij}$, $i = 1, 2, \ldots, m$; $j = 1, 2, \ldots, n$; $\widetilde{z}_i = tz_i$, $i = 0, 1, \ldots, m$), le problème de programmation stochastique (3.24)-(3.26) s'écrit comme un problème déterministe de programmation convexe,

$$\widetilde{z}_0 \to \max \tag{3.33}$$

sous les conditions

$$\overline{C}\widetilde{D}\overline{B} - \widetilde{z}_0 \geqslant t\,(C^\circ,\ X^\circ), \quad -\widetilde{V}(\widetilde{D}) + 1 \geqslant 0,$$
$$\widetilde{\mu}_i(\widetilde{D}) - \widetilde{z}_i \geqslant 0, \quad -k_i^2\,[\widetilde{\sigma}_i^2(\widetilde{D}) - \widetilde{\mu}_i^2(\widetilde{D})] + \widetilde{z}_i^2 \geqslant 0,$$
$$t \geqslant 0, \quad \widetilde{z}_i \geqslant 0, \quad i = 1, 2, \ldots, m.$$

Toute contrainte du problème découpe dans l'espace des variables $\widetilde{d}_{ij}$, $\widetilde{z}_i$, t un ensemble convexe. L'intersection de tous ces ensembles est aussi un ensemble convexe.

Ainsi, sous la condition supplémentaire $X = DB$, les problèmes de programmation stochastique des modèles de type E, V et P aux contraintes probabilistes se ramènent à des problèmes déterministes de programmation convexe.

3.4. En conclusion de ce paragraphe, considérons le problème de planification à long terme qui est un E-modèle aux contraintes probabilistes [78].

Ce problème consiste à calculer les volumes x_j de production d'un produit homogène en n périodes ($j = 1, \ldots, n$) de façon à minimiser l'espérance mathématique du coût total de production et de stockage du produit, tout en respectant les deux conditions suivantes :

a) satisfaire avec une probabilité au moins égale à $p_j^{(1)}$ la demande b_j ($j = 1, 2, \ldots, n$) à distribution normale aléatoire ;

b) au début de la j-ième période, le stock y_j de produit ne doit pas sortir des limites $y_{\min}$ et $y_{\max}$ données d'avance, et ce avec une probabilité au moins égale à $p_j^{(2)}$.

Les conditions a) et b) peuvent être mises sous la forme

$$P\,(x_j + y_j - b_j \geqslant 0) \geqslant p_j^{(1)}, \quad j = 1, 2, \ldots, n. \tag{3.34}$$
$$P\,(y_{\min} \leqslant x_j + y_j - b_j \leqslant y_{\max}) \geqslant p_j^{(2)}, \quad j = 1, 2, \ldots, n. \tag{3.35}$$
$$x_j \geqslant 0, \quad y_j \geqslant 0, \quad j = 1, 2, \ldots, n. \tag{3.36}$$

Au début de la j-ième période, le stock y_j est lié au volume de production, au stock et à la demande de la période précédente par les relations évidentes suivantes:

$$y_j = \begin{cases} x_{j-1} + y_{j-1} - b_{j-1} & \text{si} \quad x_{j-1} + y_{j-1} - b_{j-1} \geqslant 0, \\ 0 \quad \text{si} \quad x_{j-1} + y_{j-1} - b_{j-1} < 0, & j = 1, 2, \ldots, n. \end{cases}$$

$$x_0 = 0, \quad b_0 = 0, \quad y_0 \text{ donné.}$$

D'où

$$y_j \geqslant \begin{cases} x_{j-1} + y_{j-1} - b_{j-1}, \\ 0, \quad j = 1, 2, \ldots, n, \end{cases} \tag{3.37}$$

$$x_0 = 0, \quad b_0 = 0, \quad y_0 \text{ donné.}$$

Désignons respectivement par $c_j^{(1)}$ et $c_j^{(2)}$ les coûts de fabrication et de stockage d'une unité de produit au cours de la j-ième période. Le vecteur du coût peut être aléatoire. Le coût de stockage peut dépendre, par exemple, des conditions atmosphériques.

En négligeant le coût relatif à la variation des stocks, on peut écrire le facteur de qualité d'un programme (espérance mathématique du coût total relatif à toutes les n périodes) sous la forme

$$E\left\{\sum_{j=1}^{n} (c_j^{(1)} x_j + c_j^{(2)} y_j)\right\}. \tag{3.38}$$

Le problème de planification de la production et du stockage d'un produit homogène se ramène ainsi au calcul des quantités x_j et y_j telles qu'elles minimisent le coût (3.38) sous les conditions (3.34)-(3.37).

La décision relative à x_j est prise au début de la j-ième période d'après l'information sur le chiffre de la demande aux périodes précédentes. C'est pourquoi le volume de production x_j est une fonction des grandeurs aléatoires $b_1, b_2, \ldots, b_{j-1}$. Posons que cette dépendance est linéaire:

$$x_j = d_{j0} + \sum_{i=1}^{j-1} d_{ji} b_i. \tag{3.39}$$

La contrainte additionnelle naturelle (3.39) ramène le problème de planification de la production au type (3.5)-(3.7). La matrice D, dans ce problème, est par hypothèse triangulaire ($d_{ji} = 0$ pour $i \geqslant j$). Le choix des valeurs d_{ji} qui minimisent le coût total de fabrication et de stockage du produit sous l'hypothèse de satisfaire à la demande et de garantir avec les probabilités imposées les limites assignées au stock se ramène, conformément aux considérations du paragraphe 3.2, à un problème de programmation convexe.

Les conditions concrètes de la planification de la production peuvent imposer le remplacement du E-modèle considéré par un V-modèle ou un P-modèle.

Dans les cas où la planification intéresse des entreprises différentes et où il importe d'assurer une répartition égale (des frais et, par conséquent, des revenus) entre ces entreprises, il peut s'avérer utile de remplacer le E-modèle par un V-modèle. Le facteur de qualité (3.38) doit alors être remplacé par

$$E\,[(C,\ X) - (C^\circ,\ X^\circ)]^2,$$

où la quantité $(C^\circ,\ X^\circ)$ est définie par le prix de revient prévu de fabrication et de stockage du produit.

Dans les cas où il importe d'assurer que les dépenses n'excèdent pas un certain niveau fixe, il est rationnel de remplacer le E-modèle par un P-modèle et, au lieu de minimiser le facteur de qualité du programme (3.38), maximiser (sous les mêmes conditions (3.34)-(3.37) et (3.39)) la probabilité

$$P\,[(C,\ X) \leqslant (C^\circ,\ X^\circ)].$$

Quel que soit le cas, la planification à long terme de la production et du stockage dans ces conditions de demande indéterminée se ramène à des problèmes de programmation convexe.

§ 4. Problèmes à deux étapes de la programmation stochastique

4.1. Considérons un problème de programmation stochastique à matrice fixe A de dimension $m \times n_1$, à vecteur donné C des coefficients de la forme linéaire de dimension n_1 et à vecteur aléatoire B des contraintes de dimension m.

On demande de calculer le vecteur X de dimension n_1 qui minimise l'espérance mathématique de la forme linéaire

$$E\,(C,\ X) \to \min \tag{4.1}$$

sous les conditions

$$AX = B, \tag{4.2}$$

$$X \geqslant 0. \tag{4.3}$$

Les relations (4.1)-(4.3) ne définissent pas encore l'énoncé du problème stochastique, puisqu'il n'y est pas spécifié ce qu'il faut comprendre par programme du problème.

Si l'on considère le problème en formulation serrée, il faut choisir la solution au sein de l'ensemble des programmes permanents, c'est-à-dire de l'ensemble des vecteurs qui vérifient les conditions du problème quel que soit l'état de la nature, c.-à-d. quelle que soit la réalisation des paramètres aléatoires du problème.

Nous avons déjà dit qu'un problème en formulation serrée peut ne pas avoir de solution : l'ensemble des programmes permanents du problème peut être vide.

Plus pratique est la formulation souple que l'on construit en partant des raisonnements suivants. On choisit d'abord un certain vecteur $X \geqslant 0$ en tant que solution préalable. Après quoi l'on observe la réalisation du vecteur aléatoire B et l'on évalue la variable d'écart $AX - B$. Au deuxième stade, on peut calculer (en conformité avec une matrice des conditions $D_{m \times n_2}$ connue d'avance) le vecteur ξ de dimension n_2 qui compense de la façon la plus économique (dans un sens donné) la variable d'écart $AX - B$. En d'autres termes, le deuxième stade est associé à la résolution du problème

$$(H, \xi) \to \min, \tag{4.4}$$

$$D\xi = B - AX, \tag{4.5}$$

$$\xi \geqslant 0. \tag{4.6}$$

On suppose que le vecteur de dimension n_2 du coût supplémentaire H et la matrice D de dimension $m \times n_2$ sont déterministes et donnés à l'avance. Ce faisant, il faut naturellement avoir soin que la solution préalable X garantisse l'existence d'un ξ satisfaisant à la condition (4.5).

Dans le cas le plus simple, D est la matrice unité, et le vecteur $\xi = \xi^+ - \xi^-$, où

$$\xi^+ = B - AX \quad \text{et} \quad \xi^- = 0 \quad \text{avec} \quad B \geqslant AX,$$

$$\xi^+ = 0 \quad \text{et} \quad \xi^- = AX - B \quad \text{avec} \quad B < AX.$$

Le vecteur ξ dépend du X choisi et est, en raison de (4.5), une fonction des composantes aléatoires du vecteur des contraintes B.

Il est logique de réunir les deux stades d'adoption d'une solution et d'énoncer le problème de la façon suivante. On demande de calculer

$$\min_X E \min_\xi [(C, X) + (H, \xi)] \tag{4.7}$$

sous les conditions

$$AX + D\xi = B, \tag{4.8}$$

$$X \geqslant 0, \quad \xi \geqslant 0. \tag{4.9}$$

Un problème de programmation stochastique de la forme (4.7)-(4.9) s'appelle problème stochastique *à deux étapes*.

De nombreux problèmes de la théorie de la gestion des stocks se ramènent à des problèmes de ce type. Les décisions préalables sur le volume de la production ou les quantités stockées doivent être prises dans des conditions d'une information incomplète sur la demande. Tel excédent de la demande sur l'offre déterminé par la suite, impose l'acquisition d'urgence (et par conséquent, à un prix plus élevé) des produits déficitaires. Lorsque la demande est insuffisante, le stockage des produits donne lieu à des frais supplémentaires.

Les résultats fondamentaux sur les problèmes à deux étapes ont été obtenus par Dantzig et Madansky et publiés dans [40].

Parmi les contraintes du type (4.8), certaines peuvent être indépendantes de ξ. Ce sont, notamment, celles qui correspondent aux composantes b_i non aléatoires du vecteur des contraintes B. Ces contraintes sont dites *fixes*.

Des contraintes additionnelles sur X peuvent être imposées aussi par la structure de la matrice D. Si, par exemple, D est une matrice non négative, un B quelconque entraîne

$$AX \leqslant B.$$

Dans le cas général, la solution préalable X doit vérifier également les conditions garantissant pour la matrice de compensation D donnée l'existence d'un vecteur correcteur $\xi \geqslant 0$ (solution du deuxième stade) qui élimine les variables d'écart dans les conditions du problème de décision.

Les contraintes de ce type s'appellent *induites*. Désignons par K l'ensemble des vecteurs $X \geqslant 0$ qui satisfont à toutes les contraintes fixes et induites, c'est-à-dire aux contraintes du problème à deux étapes qui ne contiennent pas ξ. L'ensemble K est convexe.

Nous ne considérerons par la suite que les problèmes où il existe pour tout $X \in K$ et tout B un vecteur ξ tel que le couple $[X, \xi]$ vérifie les conditions (4.8)-(4.9).

Nous verrons plus loin que dans le cas général la résolution des problèmes à deux étapes se ramène au calcul du programme optimal d'un problème de programmation convexe.

Il arrive qu'en optimisant un programme à partir des considérations quelconques on puisse profiter des critères énoncés dans les théorèmes du paragraphe suivant.

4.2. La résolution du problème (4.4)-(4.6) (ou plus exactement, le programme optimal de son dual) peut être appliquée pour construire les conditions d'optimalité du programme d'un problème à deux étapes. Dans le cas où le nombre de réalisations possibles du vecteur aléatoire B est infini, les conditions d'optimalité du programme d'un problème à deux étapes stochastique présentent surtout un intérêt théorique. Les problèmes de programmation convexe auxquels se ramènent dans ce cas la résolution des problèmes à deux étapes sont assez complexes et de discussion laborieuse. Lorsque l'ensemble des réalisations du vecteur des contraintes B est fini, les conditions d'optimalité d'un programme définissent la procédure de résolution du problème à deux étapes. La détermination du programme optimal du problème (4.7)-(4.9) est ramenée dans ce cas à la programmation par blocs.

Pour condenser l'exposé, appelons (4.1)-(4.3) problème I et (4.4)-(4.6), problème II.

Considérons le problème $\widetilde{\text{II}}$, dual du problème II pour B et X fixes. Il consiste à calculer le vecteur η maximisant la forme linéaire

$$\eta\,(B - AX) \tag{4.10}$$

sous les conditions

$$\eta D \leqslant H. \tag{4.11}$$

Supposons que le problème $\widetilde{\text{II}}$ est résoluble quels que soient B et $X \in K$.

Si, notamment, l'ensemble K n'est pas vide, le problème $\widetilde{\text{II}}$ admet une solution.

Désignons par $\eta^* = \eta^*(B, X)$ le programme optimal du problème $\widetilde{\text{II}}$. Si le problème $\widetilde{\text{II}}$ a plus d'une solution, nous entendrons par vecteur η^* l'une quelconque d'entre elles.

Du premier théorème de dualité il résulte que

$$\min_{\xi}(H, \xi) = \eta^*(B, X)(B - AX). \tag{4.12}$$

Désignons par $C(B, X)$ l'expression

$$C(B, X) = \min_{\xi}[(C, X) + (H, \xi)] = (C, X) + \eta^*(B, X)(B - AX). \tag{4.13}$$

Dans les notations adoptées, il faut comprendre par solution du problème à deux étapes (4.7)-(4.9) le vecteur $X \in K$ qui minimise $EC(B, X)$.

Voici quelques propositions qui facilitent pour le moins la discussion qualitative des problèmes à deux étapes de programmation stochastique [40].

On montre sans peine (cf. exercice 8) que $C(B, X)$ est convexe vers le bas sur X quel que soit B et que, par conséquent, $EC(B, X)$ est une fonction convexe vers le bas du point $X = (x_1, x_2, \ldots, x_n)$.

Ainsi, la formulation du problème à deux étapes conduit à des problèmes de programmation convexe.

Introduisons le vecteur

$$C_{X_1} = C - E\eta^*(B, X_1)\,A$$

et la forme linéaire

$$L_{X_1}(X) = (C_{X_1}, X) = [C - E\eta^*(B, X_1)\,A]\,X.$$

Cette écriture donne lieu au

Théorème 4.1. (Condition nécessaire d'optimalité du programme d'un problème à deux étapes.

Si X^* est solution d'un problème à deux étapes, alors pour tout $X_1 \in K$

$$L_{X_1}(X^*) \leqslant L_{X_1}(X_1). \tag{4.14}$$

Démonstration. Etant donné que X^* est un programme optimal et X_1 un programme du problème à deux étapes, il vient

$$EC\,(B,\ X^*) \leqslant EC\,(B,\ X_1),$$

ou, ce qui revient au même,

$$(C,\ X^*) + E\eta^*\,(B,\ X^*)\,(B - AX^*) \leqslant$$
$$\leqslant (C,\ X_1) + E\eta^*\,(B,\ X_1)\,(B - AX_1). \qquad (4.15)$$

En outre,

$$E\eta^*\,(B,\ X^*)\,(B - AX^*) \geqslant E\eta^*\,(B,\ X_1)\,(B - AX^*), \qquad (4.16)$$

puisque $\eta^*\,(B,\ X^*)$ est un programme optimal du problème $\widetilde{\Pi}$ avec $X = X^*$. En retranchant (4.16) de (4.15), on obtient la proposition (4.14).

Le théorème 4.1 définit l'idée de l'analyse du problème à deux étapes. Il affirme que la solution X^* de ce problème donne à la forme linéaire

$$L_{X_1}(X) = [C - E\eta^*\,(B,\ X_1)]\,X$$

une valeur qui ne dépasse pas celle de la forme linéaire au point $X_1 \in K$. D'où la méthode suivante de discussion du problème (4.7)-(4.9). On choisit un certain nombre de points $X_1 \in K$ et on calcule pour eux la solution $\eta^*\,(B,\ X_1)$ du problème $\widetilde{\Pi}$. Pour chaque X_1 choisi on construit une inégalité du type (4.14). La suite d'inégalités obtenue est un système de contraintes qui restreignent l'ensemble contenant l'optimum et diminuent, par conséquent, la marge de variation du facteur de qualité de la solution du problème à deux étapes.

Théorème 4.2. (Condition nécessaire et suffisante d'optimalité du programme d'un problème à deux étapes.)

Soit X^* un point intérieur de l'ensemble K, et $EC\,(B,\ X)$ une fonction dérivable dans le voisinage de X^*. Le problème $\widetilde{\Pi}$ a alors une solution $\eta^*\,(B,\ X^*)$ telle que

$$C_{X^*} = C - E\eta^*\,(B,\ X^*)\,A = 0 \qquad (4.17)$$

si et seulement si X^* est solution du problème à deux étapes.

Démonstration. Constatons d'abord que l'hyperplan

$$z = [C - E\eta^*\,(B,\ X_1)\,A]\,X + E\eta^*\,(B,\ X_1)\,B \qquad (4.18)$$

est hyperplan d'appui de l'ensemble $z \geqslant EC\,(B,\ X)$ au point $X = X_1$. En effet, en $X = X_1$, cet hyperplan coupe l'hypersurface $z = EC\,(B,\ X)$. En outre, comme $\eta^*\,(B,\ X)$ est un programme

optimal du problème $\widetilde{\mathrm{II}}$, alors

$$E\eta^* (B, X) (B - AX) \geqslant E\eta^* (B, X_1) (B - AX).$$

C'est pourquoi

$$[C - E\eta^* (B, X_1) A] X + E\eta^* (B, X_1) B \leqslant$$
$$\leqslant [C - E\eta^* (B, X) A] X + E\eta^* (B, X) B = EC (B, X).$$

Cette inégalité confirme que l'hyperplan (4.18) est hyperplan d'appui de l'ensemble convexe $z \geqslant EC (B, X)$.

Par hypothèse, la fonction convexe $EC (B, X)$ est dérivable en $X = X^*$. Par suite, l'hyperplan d'appui

$$z = [C - E\eta^* (B, X^*) A] X + E\eta^* (B, X^*) B$$

est tangent en $X = X^*$ à l'hypersurface $z = EC (B, X)$.

Compte tenu de ce que X^* est un point intérieur de l'ensemble K, on en tire que l'égalité

$$\frac{\partial EC (B, X)}{\partial X}\bigg|_{X=X^*} = \frac{\partial z}{\partial X}\bigg|_{X=X^*} = C - E\eta^* (B, X^*) A = 0 \quad (4.19)$$

constitue une condition nécessaire d'optimalité du vecteur X^*.

L'égalité (4.17) est également une condition suffisante, puisque la fonction $C (B, X)$, comme on l'a signalé plus haut, est convexe vers le bas.

Le théorème est ainsi démontré.

Prêtons attention au fait que, comme il est possible de le montrer, la fonction $EC (B, X)$, dans le cas d'un nombre fini de réalisations du vecteur aléatoire des contraintes B, ne justifie pas les conditions du théorème 4.2: en effet, elle n'est pas dérivable en X^*.

4.3. Dans le cas général, où le nombre de réalisations du vecteur aléatoire B est infini, la résolution d'un problème à deux étapes peut se ramener à celle d'un problème de programmation convexe.

Supposons que l'ensemble K (ensemble des vecteurs $X \geqslant 0$ qui vérifient toutes les contraintes fixes et induites) soit un ensemble convexe

$$A^\circ X = B^\circ,$$
$$X \geqslant 0.$$

Un problème à deux étapes peut être énoncé, d'après les propositions du paragraphe 4.2, de la façon suivante. Soit à calculer le vecteur X qui satisfait aux conditions

$$A^\circ X = B^\circ,$$
$$EC (B, X) \leqslant W,$$
$$X \geqslant 0$$

et minimise W.

Pour résoudre ce problème de programmation convexe, Kelley [45] a proposé la méthode itérative suivante. Supposons que le point X_0 soit un programme initial du problème à deux étapes et $\eta^*(B, X)$, comme auparavant, un programme optimal du problème $\widetilde{\mathrm{II}}$. Considérons le problème linéaire suivant:

$$W \to \min,$$
$$(C, X) + E\eta^*(B, X_0)(B - AX) \leqslant W,$$
$$A^\circ X = B^\circ,$$
$$X \geqslant 0.$$

Soit X_1 le programme optimal de ce problème. L'itération suivante de la résolution du problème de programmation convexe auquel on a réduit le problème à deux étapes est un nouveau problème linéaire

$$W \to \min,$$
$$(C, X) + E\eta^*(B, X_0)(B - AX) \leqslant W,$$
$$(C, X) + E\eta^*(B, X_1)(B - AX) \leqslant W,$$
$$X \geqslant 0, \quad X \in K.$$

A la k-ième itération, on aboutit au problème de programmation linéaire

$$W \to \min,$$
$$(C, X) + E\eta^*(B, X_i)(B - AX) \leqslant W, \quad i = 0, 1, \ldots, k-1,$$
$$X \geqslant 0, \quad X \in K.$$

Kelley a montré que

$$\lim_{k\to\infty} EC(B, X_k) = \min_{X\in K} EC(B, X).$$

4.4. Dans [40] on a indiqué la possibilité d'utiliser la méthode de décomposition pour résoudre le problème de programmation convexe, équivalent déterministe du problème à deux étapes.

Considérons le cas d'un nombre fini de réalisations du vecteur aléatoire B. Supposons que le vecteur des contraintes B puisse prendre un nombre fini de valeurs $B^{(1)}, B^{(2)}, \ldots, B^{(N)}$ conformément aux probabilités $p^{(1)}, p^{(2)}, \ldots, p^{(N)}$ $\left(\sum_{j=1}^{N} p^{(j)} = 1\right)$. Voici l'énoncé qu'on peut alors donner au problème à deux étapes.

Calculer les vecteurs $X, \xi^{(1)}, \ldots, \xi^{(N)}$ minimisant la forme linéaire

$$L = (C, X) + \sum_{j=1}^{N} p^{(j)}(H, \xi^{(j)}) \tag{4.20}$$

sous les conditions

$$\left.\begin{array}{l} AX + D\xi^{(1)} \qquad\qquad\qquad\qquad = B^{(1)}, \\ AX \qquad\quad + D\xi^{(2)} \qquad\qquad = B^{(2)}, \\ \cdots\cdots\cdots\cdots\cdots\cdots \\ AX \qquad\qquad\qquad\quad + D\xi^{(N)} = B^{(N)}, \end{array}\right\} \tag{4.21}$$

$$X \geqslant 0, \quad \xi^{(j)} \geqslant 0, \quad j = 1, 2, \ldots, N. \tag{4.22}$$

Dans [40] on démontre les conditions nécessaires et suffisantes d'optimalité du programme d'un problème à deux étapes ayant un nombre fini de réalisations du vecteur aléatoire des contraintes.

Théorème 4.3. (Condition suffisante d'optimalité d'un programme.)

Si, pour $X = X^*$, il existe une solution η^* (B, X^*) du problème $\widetilde{II}$ telle que

$$C_{X^*} = C - E\eta^* (B, X^*) A \geqslant 0, \tag{4.23}$$

$$L_{X^*} (X^*) = [C - E\eta^* (B, X^*) A] X^* = 0. \tag{4.24}$$

X^* est alors solution du problème à deux étapes.

Théorème 4.4. (Condition nécessaire d'optimalité d'un programme.)

Si X^* est un programme optimal du problème à deux étapes, il existe alors une solution du problème $\widetilde{II}$ qui vérifie les relations (4.23)-(4.24).

Il est logique de réunir les théorèmes 4.3 et 4.4 en un théorème 4.5 qui établit une condition nécessaire et suffisante d'optimalité du programme.

Théorème 4.5. *Pour optimer le programme X^* d'un problème à deux étapes il faut et il suffit qu'il existe pour $X = X^*$ une solution η^* (B, X^*) du problème $\widetilde{II}$ vérifiant les relations* (4.23)-(4.24).

Le théorème 4.5 précise le théorème 4.2 pour un nombre fini de réalisations du vecteur B, cas où les conditions du théorème 4.2 sont compromises.

Ecrivons le problème dual du problème (4.20)-(4.22). (On prend ici comme variables du dual les paramètres $p^{(i)}\eta^{(i)}$.)

Maximiser la forme linéaire

$$p^{(1)}B^{(1)}\eta^{(1)} + p^{(2)}B^{(2)}\eta^{(2)} + \ldots + p^{(N)} B^{(N)}\eta^{(N)} \tag{4.25}$$

sous les conditions

$$p^{(1)}A^T\eta^{(1)} + p^{(2)}A^T\eta^{(2)} + \ldots + p^{(N)}A^T\eta^{(N)} \leqslant C, \tag{4.26}$$

$$\left.\begin{array}{l} D^T\eta^{(1)} \leqslant H, \\ \quad D^T\eta^{(2)} \leqslant H, \\ \ldots\ldots\ldots \\ \qquad D^T\eta^{(N)} \leqslant H. \end{array}\right\} \tag{4.27}$$

La structure du problème (4.25)-(4.27) permet, en utilisant les méthodes de programmation par blocs (cf. chapitre 4), de définir le programme optimal X^* du problème à deux étapes.

En résolvant le problème (4.25)-(4.27) par la méthode de décomposition, il est rationnel de prendre les conditions (4.26) comme contraintes d'« enchaînement » et de considérer les conditions (4.27) comme un bloc unique. Le Z-problème correspondant compte alors $n_1 + 1$ conditions. Avec les notations du chapitre 4, le Z-problème correspondant est de la forme

$$\sum_{\nu=1}^{s} \sigma_\nu z_\nu \to \max \tag{4.28}$$

sous les conditions

$$\sum_{\nu=1}^{s} P_\nu z_\nu \leqslant C, \tag{4.29}$$

$$\sum_{\nu=1}^{s} z_\nu = 1, \tag{4.30}$$

$$z_\nu \geqslant 0, \quad \nu = 1, 2, \ldots, s. \tag{4.31}$$

(Nous posons ici, pour simplifier, que le domaine défini par les conditions (4.27) est borné.)

Il est clair que le vecteur $\Lambda^{(1)}$ de dimension n_1 des critères des conditions (4.29) par rapport à la base optimale du Z-problème se confond avec le programme optimal X^* du problème à deux étapes de programmation stochastique.

Nous laissons au lecteur (cf. exercice 11) le soin de démontrer que le critère d'optimalité de la résolution du problème (4.25)-(4.27) par la méthode de décomposition coïncide avec la condition suffisante d'optimalité d'un programme du problème à deux étapes, énoncée dans le théorème 4.3.

§ 4.5. Considérons quelques modèles dont l'analyse se ramène à la résolution de problèmes à deux étapes.

Soit X le vecteur qui définit les volumes de production des produits finis; A la matrice technologique, C le vecteur du coût; B le vecteur de la demande du produit fini. On suppose que la matrice A et le vecteur C soient fixés et que le vecteur B soit aléatoire. Dési-

gnons par $P_i(b_i)$ la densité de distribution de la i-ème composante du vecteur des contraintes.

Si $X^{(0)}$ est la décision adoptée sur le volume de production des produits intermédiaires et B la demande qui se manifeste, le coût total se calcule alors d'après la formule

$$(C, X^{(0)}) + \sum_{i=1}^{m} h_i \max\left(0, b_i - \sum_{j=1}^{n} a_{ij} x_j^{(0)}\right).$$

Le premier terme $(C, X^{(0)})$ définit le coût de production direct et le deuxième, la pénalité pour l'insatisfaction de la demande (h_i la pénalité pour l'insatisfaction de la demande par unité de produit du i-ème type). Le vecteur $H = (h_1, \ldots, h_m)$ peut être interprété autrement qu'une pénalité. Les frais dus à l'insatisfaction de la demande peuvent intervenir, par exemple, par suite de la nécessité de compenser les produits déficients par d'autres produits plus chers.

Le problème suppose que le coût de stockage d'une unité de produit excédentaire est nettement inférieur à celui de la production et de la pénalité pour l'insatisfaction de la demande.

Le problème peut être mis sous la forme suivante. Minimiser l'espérance mathématique du coût

$$E\,[(C, X) + (H, \xi^+)] \to \min \tag{4.32}$$

sous les conditions

$$AX + \xi^+ - \xi^- = B, \tag{4.33}$$

$$X \geqslant 0, \quad \xi^+ \geqslant 0, \quad \xi^- \geqslant 0. \tag{4.34}$$

Ici

$$H = (h_1, \ldots, h_m); \quad \xi^+ = (\xi_1^+, \ldots, \xi_m^+); \quad \xi^- = (\xi_1^-, \ldots, \xi_m^-);$$
$$\xi^+ = B - AX, \quad \xi^- = 0 \quad \text{pour} \quad B \geqslant AX,$$
$$\xi^- = AX - B, \quad \xi^+ = 0 \quad \text{pour} \quad B < AX.$$

C'est un problème à deux étapes de programmation stochastique.

Introduisons les variables u_i:

$$u_i = \sum_{j=1}^{n} a_{ij} x_j, \quad i = 1, 2, \ldots, m. \tag{4.35}$$

Le problème (4.32)-(4.34) peut être alors énoncé de la façon suivante.

Calculer les variables $x_j \geqslant 0$ $(j = 1, 2, \ldots, n)$ et u $(i = 1, 2, \ldots, m)$ minimisant la fonction économique

$$\sum_{j=1}^{n} c_j x_j + \sum_{i=1}^{m} h_i \int_{u_i}^{\infty} P_i(b_i)(b_i - u_i)\, db_i \tag{4.36}$$

sous les conditions

$$\sum_{j=1}^{n} a_{ij}x_j - u_i = 0, \quad i = 1, 2, \ldots, m; \tag{4.37}$$

$$x_j \geqslant 0, \quad j = 1, 2, \ldots, n.$$

Le problème stochastique à deux étapes est ramené à un problème de programmation convexe (cf. exercice 12).

Il existe des artifices qui permettent dans plusieurs cas de simplifier l'exploration des problèmes de programmation stochastique. Considérons à titre d'exemple le problème de distribution à demande indéterminée [70].

Le problème de distribution déterministe consiste (cf. [87], chapitre 3, § 2) à minimiser la forme linéaire

$$\sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij}x_{ij} \tag{4.38}$$

sous les conditions

$$\sum_{j=1}^{n} x_{ij} = a_i, \quad i = 1, 2, \ldots, m, \tag{4.39}$$

$$\sum_{i=1}^{m} k_{ij}x_{ij} = b_j, \quad j = 1, 2, \ldots, n; \tag{4.40}$$

$$x_{ij} \geqslant 0, \quad i = 1, 2, \ldots, m; \quad j = 1, 2, \ldots, n. \tag{4.41}$$

Supposons maintenant que la demande b_j soit aléatoire et que, de plus, le nombre de ses réalisations soit fini : la demande du j-ième produit peut prendre s_j valeurs $b_{j1}, b_{j2}, \ldots, b_{js_j}$ avec les probabilités $p_{j1}, p_{j2}, \ldots, p_{js_j}$. Supposons que pour tout j les réalisations b_{js} aient été numérotées dans l'ordre de leurs valeurs croissantes.

Désignons par Δ_{jt} les échelons d'accroissement de la demande :

$$\Delta_{jt} = b_{jt} - b_{j,t-1}, \quad (\Delta_{j0} = 0),$$

et par q_{jt} la probabilité d'une demande supérieure à b_{jt} :

$$q_{jt} = 1 - \sum_{l=1}^{t} p_{jl}.$$

La probabilité q_{js} est monotone décroissante avec l'augmentation de s.

Soient h_j la pénalité pour l'insatisfaction de la demande par unité du j-ième produit et ξ_j^+ la grandeur de la demande insatisfaite du j-ième produit.

Les dépenses totales, constituées des frais de transport et de la pénalité pour l'insatisfaction de la demande, sont égales à

$$\sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij}+\sum_{j=1}^{n} h_j\xi_j^+. \tag{4.42}$$

Le problème de distribution stochastique se ramène ainsi à minimiser l'espérance mathématique (4.42) sous les conditions:

$$\sum_{i=1}^{m} k_{ij}x_{ij}+\xi_j^+-\xi_j^-=b_j, \quad j=1, 2, \ldots, n, \tag{4.43}$$

$$\sum_{j=1}^{n} x_{ij}=a_i, \quad i=1, 2, \ldots, m, \tag{4.44}$$

$$x_{ij}\geqslant 0, \quad i=1, 2, \ldots, m; \quad j=1, 2, \ldots, n, \tag{4.45}$$

$$\xi_j^+\geqslant 0, \ \xi_j^-\geqslant 0, \quad j=1, 2, \ldots, n. \tag{4.46}$$

Ici

$$\xi_j^+=b_j-\sum_{i=1}^{m} k_{ij}x_{ij}, \ \xi_j^-=0 \text{ avec } b_j\geqslant\sum_{i=1}^{m} k_{ij}x_{ij},$$

$$\xi_j^-=\sum_{i=1}^{m} k_{ij}x_{ij}-b_j, \ \xi_j^+=0 \text{ avec } b_j<\sum_{i=1}^{m} k_{ij}x_{ij}.$$

Montrons que la minimisation de l'espérance mathématique (4.42) sous les conditions (4.43)-(4.46) se réduit à résoudre le problème spécial déterministe suivant de programmation linéaire.

Calculer les variables x_{ij} et ξ_{ij} ($i = 1, 2, \ldots, m$; $j = 1, 2, \ldots, n$), minimisant la forme linéaire

$$\sum_{i=1}^{m}\sum_{j=1}^{n} c_{ij}x_{ij}-\sum_{j=1}^{n} h_j\sum_{s=1}^{s_j} q_{js}\xi_{js} \tag{4.47}$$

sous les conditions

$$\sum_{i=1}^{m} k_{ij}x_{ij}=\sum_{s=1}^{s_j}\xi_{js}, \quad j=1, 2, \ldots, n, \tag{4.48}$$

$$\sum_{j=1}^{n} x_{ij}=a_i, \quad i=1, 2, \ldots, m, \tag{4.49}$$

$$x_{ij}\geqslant 0, \quad i=1, 2, \ldots, m; \quad j=1, 2, \ldots, n; \tag{4.50}$$

$$0\leqslant\xi_{js}\leqslant\Delta_{js}, \quad s=1, 2, \ldots, s_j; \quad j=1, 2, \ldots, n. \tag{4.51}$$

Les probabilités q_{js}, $s<s_j$ sont positives. En outre, les q_{js} sont monotones décroissantes avec l'augmentation de s. Cela traduit le fait que si la solution du problème (4.47)-(4.50) compte une composante $\xi_{jr}\neq 0$, alors avec $s<r$ tous les $\xi_{js}=\Delta_{js}$.

Désignons par s_j° le plus grand indice tel que la composante ξ_{js} de la solution du problème (4.47)-(4.51) soit différente de zéro.

Les conditions (4.48) du problème déterministe (4.47)-(4.51) et les conditions (4.43) du problème stochastique initial, qui consiste à minimiser l'espérance mathématique (4.42) sous les conditions (4.43)-(4.46), peuvent alors s'écrire

$$\sum_{i=1}^{m} k_{ij}x_{ij} = \sum_{s=1}^{s_j^\circ} \xi_{js}, \quad j=1, 2, \ldots, n. \tag{4.52}$$

et l'espérance mathématique du coût (4.42) peut être représentée comme la forme linéaire

$$L = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij}x_{ij} + \sum_{j=1}^{n} h_j \sum_{s=s_j^\circ}^{s_j} q_{js}\tilde{\xi}_{js}, \tag{4.53}$$

où

$$\tilde{\xi}_{js_j^\circ} = \Delta_{js_j^\circ} - \xi_{js_j^\circ}; \quad \tilde{\xi}_{js} = \Delta_{js} \text{ avec } s > s_j^\circ.$$

(Pour $s < s_j^\circ$, la pénalité n'est pas payée puisque dans ce cas, en vertu de (4.52), la demande est satisfaite.)

Transformons la fonction économique (4.53) en y ajoutant et retranchant la somme

$$\sum_{j=1}^{n} h_j \sum_{s=1}^{s_j^\circ} q_{js}\xi_{js}.$$

Il vient

$$L = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij}x_{ij} + \sum_{j=1}^{n} h_j \sum_{s=1}^{s_j} q_{js}\Delta_{js} - \sum_{j=1}^{n} h_j \sum_{s=1}^{s_j^\circ} q_{js}\xi_{js}.$$

La quantité

$$L_0 = \sum_{j=1}^{n} h_j \sum_{s=1}^{s_j} q_{js}\Delta_{js}$$

est l'espérance mathématique de la pénalité pour ignorance totale de la demande, lorsque aucune unité de produit n'est livrée aux points de consommation. La valeur L_0 ne dépend pas des paramètres recherchés. En outre,

$$\sum_{j=1}^{n} h_j \sum_{s=1}^{s_j^\circ} q_{js}\xi_{js} = \sum_{j=1}^{n} h_j \sum_{s=1}^{s_j} q_{js}\xi_{js}$$

du fait que $\xi_{js} = 0$ avec $s > s_j^\circ$.

Cela signifie que l'expression (4.53) de l'espérance mathématique du coût du problème stochastique initial ne diffère que par la constante L_0 de la fonction économique (4.47) du problème déterministe (4.47)-(4.51). Nous avons démontré ainsi que le problème à deux

étapes qui consiste à minimiser (4.42) sous les conditions (4.43)-(4.46) est équivalent au problème déterministe spécial de programmation linéaire.

§ 5. Evaluation des solutions approchées

La résolution des problèmes de programmation stochastique nécessite des calculs très délicats qui posent devant nous les questions suivantes:

1. Dans quels cas est-il possible d'obtenir la résolution d'un problème stochastique en remplaçant les paramètres aléatoires par leurs espérances mathématiques?

2. Comment évaluer les solutions d'un problème de programmation stochastique lorsqu'on a à sa disposition les programmes optimaux de tels ou tels problèmes de programmation mathématique engendrés par ce problème stochastique?

Pareilles questions ont été étudiées dans [11] et [49]. Nous énoncerons les résultats essentiels obtenus dans ces articles.

Soit $f(X, q)$ la valeur du facteur de qualité du programme X d'un problème stochastique sous la condition que l'état de la nature est q, et que $P(q)$ est la densité de distribution des états de la nature q. Le problème de programmation stochastique énoncé en ces termes consiste à calculer le vecteur X sur lequel on obtient l'extrémum

$$F(X) = \int_q f(X, q) P(q) dq. \tag{5.1}$$

Le théorème suivant est immédiat.

Théorème 5.1. *Si la fonction $f(X, q)$ est divisible, c'est-à-dire si elle peut être mise sous la forme*

$$f(X, q) = \sum_{\mu=1}^{s} f_\mu^{(1)}(X) f_\mu^{(2)}(q), \tag{5.2}$$

la minimisation de $F(X)$ se ramène à celle de la somme

$$\sum_{\mu=1}^{s} \lambda_\mu f_\mu^{(1)}(X), \tag{5.3}$$

où

$$\lambda_\mu = \int_q f_\mu^{(2)}(q) P(q) dq. \tag{5.4}$$

La proposition qui suit est moins évidente, mais plus utile pour les applications [49].

Théorème 5.2. *Pour rendre réalisable la solution d'un problème à deux étapes à vecteur des contraintes B aléatoire en remplaçant la demande aléatoire B par son espérance mathématique, il suffit de mettre la fonction C (B, X) sous la forme*

$$C(B, X) = C_1(X) + C_2(B) + C_3(X)B. \tag{5.5}$$

Rappelons que, d'après (4.13),

$$C(B, X) = (C, X) + \eta^*(B, X)(B - AX).$$

Considérons comme auparavant les problèmes stochastiques à vecteur des contraintes B aléatoire.

Désignons par C^* la valeur optimale du facteur de qualité d'un programme du problème à deux étapes:

$$C^* = \min_{X \in K} EC(B, X). \tag{5.6}$$

Faisons correspondre au problème à deux étapes le problème déterministe suivant de programmation linéaire. Calculer

$$\min (C, X) \tag{5.7}$$

sous les conditions

$$AX = EB, \tag{5.8}$$

$$X \geqslant 0, \tag{5.9}$$

où EB, comme d'habitude, est l'espérance mathématique du vecteur aléatoire B.

Désignons par C_1^* la valeur optimale de la forme linéaire du problème (5.7)-(5.9):

$$C_1^* = \min_X C(EB, X). \tag{5.10}$$

Associons au problème de programmation stochastique un autre problème déterministe de programmation linéaire:

$$(C, X) \to \min \tag{5.11}$$

sous les conditions

$$AX = B, \tag{5.12}$$

$$X \geqslant 0, \tag{5.13}$$

où B est une réalisation fixe du vecteur aléatoire des conditions.

Désignons par C_2^* l'espérance mathématique de la valeur optimale de la forme linéaire du problème (5.11)-(5.13):

$$C_2^* = E \min_X C(B, X). \tag{5.14}$$

Introduisons enfin la quantité C_3^*:

$$C_3^* = EC[B, X^*(EB)], \tag{5.15}$$

qui est la valeur du facteur de qualité du programme d'un problème stochastique dans lequel le programme optimal X^* du problème (5.7)-(5.9) est substitué au vecteur X.

Adoptons les hypothèses naturelles suivantes.

a) L'ensemble des réalisations possibles du vecteur B est convexe.

b) Les conditions (5.12)-(5.13) sont compatibles quelle que soit la réalisation du vecteur aléatoire B.

c) La pénalité définie par la variable d'écart est assez grande pour que l'information sur le vecteur B qui a été réalisé assure l'égalité $\xi = 0$ dans le programme optimal du problème à deux étapes.

Dans les hypothèses adoptées, la valeur optimale C^* du facteur de qualité d'un programme du problème stochastique est évaluée de la façon suivante, à l'aide des caractéristiques introduites C_i^* des problèmes déterministes de programmation linéaire que ce problème a engendrés.

Théorème 5.3. *Les relations*

$$C_1^* \leqslant C_2^* \leqslant C^* \leqslant C_3^* \tag{5.16}$$

ont lieu.

Les valeurs C_1^*, C_2^* et C_3^* s'obtiennent par des calculs nettement plus simples que ceux nécessaires pour résoudre un problème de programmation stochastique. Les différences $C_3^* - C_1^*$ et $C_3^* - C_2^*$ caractérisent les erreurs qui peuvent être obtenues si l'on remplace la résolution d'un problème stochastique par le calcul du programme optimal du problème de programmation linéaire dans lequel le vecteur des contraintes aléatoire B est remplacé par son espérance mathématique.

§ 6. Programmation markovienne

Parmi les modèles de programmation stochastique, les modèles probabilistes dynamiques occupent une place considérable. Leur formalisation et la construction des procédures de calcul correspondantes présentent de grandes difficultés. Les modèles correspondant aux systèmes dans lesquels la transition d'un état aléatoire en un autre est définie par une chaîne de Markov en sont une exception. Les méthodes d'étude de tels modèles font l'objet de la programmation dite markovienne. Dans [17], on montre que la discussion des modèles markoviens linéaires de problèmes extrémaux conditionnels se ramène à la résolution de problèmes de programmation linéaire. Illustrons l'idée de la programmation markovienne par un exemple suffisamment général [51].

Considérons le problème qui consiste à établir le programme de fabrication mensuel pour une période assez longue (horizon économique), que l'on peut théoriquement prendre égale à l'infini. Suppo-

sons que la demande du produit soit aléatoire, sa distribution étant discrète identique pour chaque mois. Désignons par p_n la probabilité d'une demande de n unités de produit par mois. Il est clair que dans le cas où la quantité de produit disponible dépassera la demande, il faut au mois suivant planifier la production en partant de l'excédent de produit non réalisé le mois précédent.

Le problème consiste à composer d'avance des instructions permettant de déterminer le volume de production au mois courant en fonction du volume de produit resté du mois précédent et de la demande attendue.

Le facteur de qualité de la solution du problème est l'espérance mathématique du coût total, l'horizon économique de planification étant infini.

Pour fixer les idées, rapportons au début du mois les produits non réalisés et à la fin du mois la fabrication du produit et la satisfaction de la demande.

Introduisons les notations suivantes:

i est le nombre d'unités de produit disponible au début du mois (excédent de production du mois précédent);

j, le nombre d'unités de produit à fabriquer d'après le programme en cours du mois;

n, la demande mensuelle (en unités de produit);

$k = i + j$, le stock mensuel (en unités de produit);

$s = \left\{ \begin{array}{ll} n - k & \text{avec } n > k \\ 0 & \text{avec } n \leqslant k \end{array} \right\}$, la demande insatisfaite;

$t = \left\{ \begin{array}{ll} k - n & \text{avec } k > n \\ 0 & \text{avec } k \leqslant n \end{array} \right\}$, les excédents de produit en fin de mois;

$c_1(i)$, le coût du stockage, au cours du mois, des produits non réalisés le mois précédent;

$c_2(j)$, le coût de fabrication du produit;

$c_3(s)$, la pénalité pour demande insatisfaite.

i, j, n, k, s, t sont naturellement des entiers non négatifs. Supposons également que la capacité T du magasin de stockage limite la grandeur t:

$$t \leqslant T,$$

que les capacités de production R de l'entreprise limitent la valeur j:

$$j \leqslant R;$$

et que la demande n est limitée par le nombre d'usagers N:

$$n \leqslant N.$$

Prenons comme paramètres de planification les probabilités x_{ij} que les stocks soient égaux à s unités au début du mois et que la production soit égale à j unités de produit par mois.

Les paramètres de planification ne dépendent pas du numéro du mois, du fait que nous considérons un horizon de planification infini et que seul nous intéresse le régime établi (stationnaire).

Il est clair que

$$\sum_{i,j} x_{ij} = 1, \tag{6.1}$$

$$x_{ij} \geqslant 0, \quad i = 0, 1, \ldots, T; \quad j = 0, 1, \ldots, R. \tag{6.2}$$

La valeur moyenne du coût total mensuel, facteur de qualité de la planification, s'écrit

$$EC = Ec_1(i) + Ec_2(j) + Ec_3(s).$$

D'après les notations adoptées,

$$Ec_1(i) = \sum_{i,j} x_{ij} c_1(i),$$

$$Ec_2(j) = \sum_{i,j} x_{ij} c_2(j),$$

$$Ec_3(s) = \sum_{i,j} x_{ij} \sum_{s=1}^{N-(i+j)} p_{i+j+s} c_3(s).$$

Ainsi, la fonction économique de planification est

$$EC = \sum_{i,j} x_{ij} \left[c_1(i) + c_2(j) + \sum_s p_{i+j+s} c_3(s)\right]. \tag{6.3}$$

L'état du système est défini par la quantité i (excédents de produit). La transition du système à un nouvel état est déterminée par la quantité j (volume de production prévu). Le choix de j et, par conséquent, la modification du stock dépendent seulement de l'état courant du système. Ceci signifie que la modification des stocks est un processus markovien.

Ecrivons les équations qui définissent les probabilités limites du processus de modification des stocks.

La probabilité y_i pour que l'excédent soit au début du mois égal à i s'exprime par la somme

$$y_i = \sum_j x_{ij}, \quad i = 0, 1, \ldots, T. \tag{6.4}$$

La probabilité y_t pour que l'excédent de produits soit à la fin du mois égal à t est donnée par l'expression

$$\left.\begin{aligned} y_t &= \sum_{i+j-n=t} p_n x_{ij}, \quad t = 1, 2, \ldots, T, \\ y_0 &= \sum_{i+j-n \leqslant 0} p_n x_{ij}. \end{aligned}\right\} \tag{6.5}$$

Le produit non réalisé en fin de mois constitue le stock au début du mois suivant. C'est pourquoi les formules (6.4) et (6.5) entraînent

$$\sum_j x_{0j} = \sum_{i+j-n \leqslant 0} p_n x_{ij}, \tag{6.6}$$

$$\sum_j x_{ij} = \sum_{i+j-n=t} p_n x_{ij}, \quad t = 1, 2, \ldots, T. \tag{6.7}$$

Les égalités (6.6) et (6.7) correspondent à la planification dans les conditions d'un régime stationnaire.

Nous sommes arrivés au problème suivant de programmation linéaire.

Calculer les variables x_{ij} minimisant la forme linéaire (6.3) sous les conditions (6.1), (6.2), (6.6) et (6.7).

Les variables recherchées x_{ij} définissent la probabilité conditionnelle

$$p_{ij} = \frac{x_{ij}}{\sum_j x_{ij}}$$

du fait que pour obtenir un stock fixe i il faut fabriquer j unités de produit.

La collection des nombres p_{ij} est une règle, composée d'avance, de choix de j (programme de production par mois) en fonction du stock i disponible au début du mois.

Cette formulation du problème rend possible une solution en stratégies mixtes, c'est-à-dire que la probabilité conditionnelle p_{ij} peut être un nombre quelconque dans l'intervalle [0, 1]. Cependant [65] montre que dans les problèmes de programmation markovienne, les stratégies optimales sont toujours des stratégies pures (c'est-à-dire dans la solution du problème x_{ij} est égal à zéro ou à un).

Exercices

1. Soit

$$A^{(1)} = \begin{Vmatrix} 3 & -1 \\ 5 & 2 \end{Vmatrix}; \quad A^{(2)} = \begin{Vmatrix} 0 & 2 \\ 1 & 3 \end{Vmatrix};$$

$$B^{(1)} = (5, 10)^T; \quad B^{(2)} = (2, 15)^T;$$

$$C^{(1)} = (1, 2); \quad C^{(2)} = (3, 4).$$

Résoudre le problème à une étape du type (2.1)-(2.3) dans lequel la matrice des conditions A peut coïncider avec $A^{(1)}$ avec une probabilité de 0,75 et avec $A^{(2)}$ avec une probabilité de 0,25 ; le vecteur des contraintes B prend la valeur de $B^{(1)}$ avec une probabilité de 0,9 et celle de $B^{(2)}$ avec une probabilité de 0,1 ; le vecteur de la forme linéaire C peut coïncider, avec la même probabilité, avec $C^{(1)}$ ou $C^{(2)}$.

2. Les éléments de la matrice des conditions, du vecteur des contraintes et du vecteur de la forme linéaire C sont les fonctions d'un paramètre aléatoire q

caractérisant l'état de la nature:

$$a_{ij}(q) = a_{ij}e^{(i-j)q},$$
$$b_i(q) = b_i e^{iq},$$
$$c_j(q) = c_j e^{-jq}.$$

La distribution du paramètre aléatoire q est uniforme dans l'intervalle [0, 1]. La matrice A et les vecteurs B et C sont respectivement

$$A = \begin{Vmatrix} 3 & -2 \\ 4 & 1 \end{Vmatrix}; \quad B = (2, 5)^T; \quad C = (1, -1).$$

Vérifier si le programme optimal correspondant au couple $[A(0), B(0)]$ est un programme permanent du problème (c'est-à-dire s'il satisfait aux conditions du problème pour tout $q \in [0, 1]$.

3. Etablir si la résolution par rapport aux moyennes du problème stochastique de l'exercice 2 aboutit à son programme optimal.

4. La matrice A d'un problème stochastique est fixe:

$$A = \begin{Vmatrix} 3 & -2 \\ -1 & 4 \\ -1 & 0 \\ 0 & -1 \end{Vmatrix}.$$

Le vecteur des contraintes B est un vecteur aléatoire à distribution normale, à valeur moyenne $\overline{B} = (3, 3, 0, 0)^T$ et à variance (1, 1, 1, 1). Le programme du problème est associé linéairement au vecteur B:

$$X = DB.$$

Ramener les contraintes probabilistes

$$P(AX \leqslant B)_i \geqslant 0{,}9, \quad i = 1, 2, 3, 4,$$

à la forme usuelle des conditions-inégalités.

5. Montrer que les inégalités (3.15)-(3.16) découpent dans l'espace des variables d_{ij} $(i = 1, 2, \ldots, m;\ j = 1, 2, \ldots, n)$, z_i un ensemble convexe.

6. Considérer les conditions du problème de l'exercice 4 comme les contraintes d'un E-modèle stochastique aux coefficients de la forme linéaire $c_1 = 3$ et $c_2 = 1$, pour construire un problème déterministe équivalent.

7. Considérer les conditions du problème de l'exercice 4 comme les contraintes d'un P-modèle stochastique aux coefficients de la forme linéaire $c_1 = 3$ et $c_2 = 1$ et au seuil $(C^\circ, X^\circ) = 4$, pour construire un problème de programmation convexe équivalent.

8. Démontrer que la fonction $C(B, X)$ donnée par l'expression (4.13) est convexe.

9. Résoudre le problème à deux étapes du type (4.7)-(4.9) à matrice des conditions fixe A, au vecteur de la forme linéaire C donné et au vecteur des contraintes B aléatoire.

$$A = \begin{Vmatrix} 1 & 2 & 1 \\ 3 & -2 & 4 \end{Vmatrix}; \quad C = (1, 1, 2).$$

Le vecteur B est uniformément distribué entre $B^{(1)}$ et $B^{(2)}$.

$$B^{(1)} = (5, 7)^T$$

et

$$B^{(2)} = (7, 5)^T.$$

Le vecteur correcteur ξ se calcule d'après la condition

$$D\xi = B - AX,$$

où

$$D = \begin{Vmatrix} 4 & 3 \\ 0 & 5 \end{Vmatrix}.$$

La pénalité pour correction est définie par le vecteur

$$H = (3, 4).$$

10. Les conditions étant celles du problème 9, utiliser les inégalités (5.16) pour calculer la valeur approchée $[1/2\ (C_1^* + C_3^*)]$ du minimum de la fonction économique du problème à deux étapes de programmation stochastique.

11. Montrer que le critère d'optimalité de la résolution du problème (4.25)-(4.27) par la méthode de décomposition et la condition suffisante d'optimalité du problème stochastique à deux étapes énoncée dans le théorème 4.3 sont équivalents.

12. Prouver que le problème (4.36)-(4.37) est un problème de programmation convexe.

13. Evaluer les erreurs faites sur la valeur du facteur de qualité d'un programme du problème à deux étapes de l'exercice 9 si la solution est un programme optimal du problème déterministe dans lequel le vecteur aléatoire des contraintes du problème initial est remplacé par son espérance mathématique.

14. Démontrer le théorème 5.2.

Chapitre 7

PROGRAMMATION LINÉAIRE PAR MORCEAUX

Sous le terme de programmation linéaire par morceaux on entend, dans le présent chapitre, l'ensemble des méthodes de calcul du minimum *) d'une fonction linéaire par morceaux convexe vers le bas définie sur un polyèdre convexe. La programmation linéaire par morceaux est une branche intermédiaire entre les programmations linéaire et convexe. Bien des problèmes pratiques variés peuvent être énoncés en termes de programmation linéaire par morceaux.

Le présent chapitre est consacré à la théorie et aux méthodes de résolution des problèmes de programmation linéaire par morceaux.

Dans le § 1, on donne les énoncés formels des problèmes de programmation linéaire par morceaux et on établit la liaison entre certaines classes de problèmes extrémaux conditionnels linéaires et linéaires par morceaux.

Le § 2 a pour objet les notions fondamentales et les critères d'optimalité du programme des problèmes de programmation linéaire par morceaux.

Les §§ 3 à 6 traitent avec force détails du problème particulier important de programmation linéaire par morceaux. Le facteur de qualité du programme de ce problème est une somme de fonctions linéaires par morceaux dont chacune dépend d'une seule variable, alors que ses conditions sont des égalités et inégalités linéaires. Le § 3 expose les fondements théoriques de la méthode de résolution d'un problème non dégénéré. Le § 4 lève l'hypothèse sur la non-dégénérescence des programmes du problème et étend les résultats obtenus au paragraphe précédent au cas des problèmes dégénérés.

---

*) Dans ce qui précède nous avons traité surtout des problèmes de maximisation. Dans ce qui suit nous allons considérer le problème de minimisation de la fonction économique, puisque les applications connues des modèles linéaires par morceaux relèvent du calcul du minimum du coût. Toutes les considérations rapportées dans ce chapitre se prêtent aisément à la reformulation en termes de problème de maximisation d'une fonction linéaire par morceaux convexe vers le haut, définie sur un polyèdre convexe.

L'algorithme de résolution du problème décrit au § 5 est illustré par l'exemple du § 6.

Le § 7 schématise la procédure de calcul qui permet de résoudre d'autres problèmes particuliers de programmation linéaire par morceaux.

Les méthodes de programmation linéaire par morceaux étudiées dans ce chapitre constituent une généralisation naturelle des méthodes finies de programmation linéaire. Les principes de généralisation des méthodes finies sont illustrés sur l'exemple de la méthode primale du simplexe. D'autres méthodes finies de programmation linéaire peuvent être également généralisées d'une façon analogue.

§ 1. Programmation linéaire par morceaux et problèmes linéaires

1.1. Considérons la définition formalisée d'une fonction linéaire par morceaux et d'un problème de programmation linéaire par morceaux.

Appelons dans ce chapitre fonction *linéaire par morceaux* des variables $x_1, x_2, \ldots, x_n$ la fonction $F(X)$ qui peut être mise sous la forme

$$F(X) = \max_{1 \leqslant i \leqslant s} \{L_i(X)\}, \tag{1.1}$$

où les $L_i(X) = \sum_{j=1}^{n} a_{ij}x_j - b_i$ sont des fonctions linéaires des variables $X = (x_1, x_2, \ldots, x_n)$.

On voit aisément que la fonction $F(X)$, définie par la relation (1.1), est convexe vers le bas (cf. exercice 1). Aussi serait-il plus naturel d'appeler $F(X)$ une fonction linéaire par morceaux convexe vers le bas de ses variables. Toutefois, comme nous n'aurons affaire par la suite qu'à des fonctions de ce type, garder une dénomination plus brève ne prêtera pas à confusion.

Soient $g_i(X)$ $(i = 1, 2, \ldots, m)$ les fonctions linéaires par morceaux des variables X. On voit sans peine que l'ensemble de points défini par les inégalités

$$g_i(X) \leqslant 0, \quad i = 1, 2, \ldots, m,$$

est un polyèdre convexe.

Nous entendrons par problème général de programmation linéaire par morceaux le problème qui consiste à minimiser la fonction linéaire par morceaux

$$F(X) \tag{1.2}$$

sous les conditions

$$g_i(X) \leqslant 0, \quad i = 1, 2, \ldots, m, \tag{1.3}$$

où $g_i(X)$ sont des fonctions linéaires par morceaux des variables $x_1, x_2, \ldots, x_n$.

Le système d'inégalités (1.3) est parfois remplacé par la seule inégalité vectorielle

$$G(X) \leqslant 0, \tag{1.4}$$

où $G(X) = \{g_1(X), \ldots, g_m(X)\}$ est la fonction vectorielle de l'argument vectoriel X.

Dans plusieurs cas, on aura intérêt à retirer du système de contraintes (1.3) les conditions-égalités linéaires ou autres contraintes de type spécial (non-négativité des variables, contraintes bilatérales de certaines variables, etc.). Dans de tels cas, il est logique de mettre le problème général de programmation linéaire par morceaux sous la forme

$$F(X) \to \min, \tag{1.5}$$

$$G(X) \leqslant 0, \tag{1.6}$$

$$AX - B = 0, \tag{1.7}$$

$$X \in R, \tag{1.8}$$

où R est un certain polyèdre convexe.

Voici quelques-unes des formulations particulières de problèmes de programmation linéaire par morceaux que l'on rencontre souvent dans les applications.

Problème I.

$$F(X) = \sum_{j=1}^{n} f_j(x_j) \to \min, \tag{1.9}$$

$$\sum_{j=1}^{n} a_{ij}x_j - b_i = 0, \qquad i = 1, 2, \ldots, m, \tag{1.10}$$

$$\alpha_j \leqslant x_j \leqslant \beta_j, \qquad j = 1, 2, \ldots, n, \tag{1.11}$$

où $f_j(x_j)$ est la fonction linéaire par morceaux de la variable x_j.

Problème II.

$$F(X) = \sum_{j=1}^{n} f_j(x_j) \to \min, \tag{1.12}$$

$$\sum_{j=1}^{n} g_{ij}(x_j) \leqslant 0, \qquad i = 1, 2, \ldots, m, \tag{1.13}$$

$$\alpha_j \leqslant x_j \leqslant \beta_j, \quad j = 1, 2, \ldots, n, \tag{1.14}$$

où $f_j(x_j)$ et $g_{ij}(x_j)$ sont les fonctions linéaires par morceaux de la variable x_j.

Problème III.

$$F(X)=\sum_{j=1}^{n} f_j(x_j)\to\min, \tag{1.15}$$

$$\sum_{j=1}^{n} g_{ij}(x_j)\leqslant 0, \qquad i=1, 2, \ldots, m_1, \tag{1.16}$$

$$\sum_{j=1}^{n} a_{ij}x_j-b_i=0, \qquad i=m_1+1, \ldots, m, \tag{1.17}$$

$$\alpha_j \leqslant x_j \leqslant \beta_j, \quad j = 1, 2, \ldots, n, \tag{1.18}$$

où $f_j(x_j)$ et $g_{ij}(x_j)$ sont les fonctions linéaires par morceaux de la variable x_j.

Problème IV.

$$F(X)=\sum_{k=1}^{s_0} f_k[L_{0k}(X)]\to\min, \tag{1.19}$$

$$\sum_{k=1}^{s_i} g_{ik}[L_{ik}(X)]\leqslant 0, \qquad i=1, 2, \ldots, m_1, \tag{1.20}$$

$$\sum_{j=1}^{n} a_{ij}x_j-b_i=0, \qquad i=m_1+1, \ldots, m, \tag{1.21}$$

$$\alpha_j \leqslant x_j \leqslant \beta_j, \quad j = 1, 2, \ldots, n. \tag{1.22}$$

f_k et g_{ik} sont ici les fonctions linéaires par morceaux de leurs arguments, alors que

$$L_{ik}(X)=\sum_{j=1}^{n} a_{ij}^{(k)}x_j-b_i^{(k)},$$

$$k = 1, 2, \ldots, s_i, \quad i = 0, 1, 2, \ldots, m_1,$$

sont les fonctions linéaires des variables $x_1, x_2, \ldots, x_n$.

Problème V.

$$F(X)=\sum_{j=1}^{n} f_j(x_j)+\sum_{j=n+1}^{n+s} f_j[L_j]\to\min, \tag{1.23}$$

$$\sum_{j=1}^{n} a_{ij}x_j-b_j=0, \quad i=1, 2, \ldots, m, \tag{1.24}$$

$$\alpha_j \leqslant x_j \leqslant \beta_j=0, \quad j = 1, 2, \ldots, n, \tag{1.25}$$

où f_j sont les fonctions linéaires par morceaux de leurs arguments; $L_j = L_j(X)$, les fonctions linéaires des variables $x_1, x_2, \ldots, x_n$.

Problème VI.

$$F(X)=\sum_{j=1}^{n} f_j(x_j)+\sum_{j=n+1}^{n+s} f_j[L_j]\to\min, \tag{1.26}$$

$$\sum_{j=1}^{n} g_{ij}(x_j)+\sum_{j=n+1}^{n+s} g_{ij}[L_j]\leqslant 0, \quad i=1, 2, \ldots, m, \tag{1.27}$$

$$\alpha_j \leqslant x_j \leqslant \beta_j, \quad j = 1, 2, \ldots, n, \tag{1.28}$$

où f_j et g_{ij} sont les fonctions linéaires par morceaux de leurs arguments;
$L_j = L_j(X)$, les fonctions linéaires des variables $x_1, x_2, \ldots, x_n$.

Problème VII.

$$F(X)=\sum_{j=1}^{n} f_j(x_j)+\sum_{j=n+1}^{n+s} f_j[L_j]\to\min, \tag{1.29}$$

$$\sum_{j=1}^{n} g_{ij}(x_j)+\sum_{j=n+1}^{n+s} g_{ij}[L_j]\leqslant 0, \quad i=1, 2, \ldots, m_1, \tag{1.30}$$

$$\sum_{j=1}^{n} a_{ij}x_j-b_i=0, \quad i=m_1+1, \ldots, m, \tag{1.31}$$

$$\alpha_j \leqslant x_j \leqslant \beta_j, \quad j = 1, 2, \ldots, n, \tag{1.32}$$

où f_j et g_{ij} sont les fonctions linéaires par morceaux de leurs arguments;
$L_j = L_j(X)$, les fonctions linéaires des variables $x_1, x_2, \ldots, x_n$.

On constate facilement que chacun des énoncés ci-dessus inclut certains autres comme cas particuliers. C'est pourquoi l'existence d'un algorithme de résolution de l'un d'entre eux permet de construire sans peine des schémas de calcul pour certains autres des problèmes considérés.

Les formulations des problèmes I et II se déduisent de celle du problème III avec $m_1 = 0$ et $m_1 = m$ respectivement. Le problème III, à son tour, se déduit du problème IV si pour $i = 0, 1, 2, \ldots, m_1$ on pose $s_i = n$ et

$$L_{ik}(X) = \begin{cases} x_j & \text{avec} \quad k = j, \\ 0 & \text{avec} \quad k \neq j. \end{cases}$$

Si l'on pose $L_j = \text{const}$ pour $j = n + 1, \ldots, n + s$, le problème I peut être considéré comme une formulation particulière du problème V. En introduisant les variables $x_{n+j} = L_j$, $j = 1, 2, \ldots, s$ et en posant $-\alpha_{n+j} = \beta_{n+j} = \infty$, on peut au contraire obtenir l'énoncé du problème V en tant qu'énoncé particulier du problème I.

Le problème VI inclut le problème V. La substitution $x_{n+j} = L_j$, $j = 1, 2, \ldots, s$, et les conditions $-\alpha_{n+j} = \beta_{n+j} = \infty$ permettent de considérer le problème VI, de même que le problème V, comme un cas particulier du type II.

Le problème VII inclut toutes les formulations précédentes. Mais lui aussi peut être considéré à son tour comme un problème particulier du type III.

On établit de façon analogue les relations mutuelles entre les autres couples de problèmes recensés.

1.2. On aboutit à des problèmes de programmation linéaire par morceaux à partir de considérations différentes.

De nombreux problèmes de programmation linéaire à grand nombre de variables et de contraintes comportent, en plus des conditions de type général, des contraintes spéciales. Dans plusieurs cas, on parvient à réduire de tels problèmes à des problèmes linéaires par morceaux avec un nombre nettement moindre de variables et de contraintes. L'exploration des problèmes linéaires par morceaux ainsi obtenus conduit généralement bien plus vite à la solution que la discussion de problèmes linéaires de grande taille. Le volume de calculs imposé par une itération distincte du problème linéaire par morceaux est, de façon générale, plus petit que celui exigé par une itération du problème linéaire correspondant, et le nombre d'itérations, en général, diminue lorsqu'on passe du problème linéaire au problème linéaire par morceaux.

On peut indiquer une série de problèmes techniques, économiques et bioniques qui s'énoncent directement en termes de programmation linéaire par morceaux. Ceci concerne, par exemple, certains problèmes de planification à délais imposés de la production, les problèmes de réglage automatique [32], les problèmes liés à la construction de schémas fiables à partir d'éléments qui ne le sont pas, à la synthèse de neurones formels [33], etc.

Enfin, il peut parfois être utile de réduire à un problème linéaire par morceaux certains problèmes de programmation convexe. Tout problème de programmation convexe peut être ramené de façon exacte ou approchée à un problème de programmation linéaire par morceaux. Le problème convexe en nombres entiers de la distribution des buts examiné dans [86] se ramène à un problème linéaire par morceaux équivalent. Le problème du choix d'un système de tolérances rationnel dans la construction d'appareils de précision, exposé également dans [86], peut être ramené avec l'approximation voulue à un problème de programmation linéaire par morceaux.

Ainsi, les méthodes et les algorithmes de la programmation linéaire par morceaux peuvent être utilisés à d'autres fins qu'à la résolution des problèmes linéaires par morceaux. Dans plus d'un cas, ils simplifient l'exploration des problèmes de programmation linéaire et convexe.

Dans les paragraphes qui suivent, nous examinerons quelques problèmes linéaires dont la résolution se simplifie considérablement si on les remplace par les problèmes linéaires par morceaux équivalents.

1.3. Certains problèmes de planification (cf., par exemple, [89]) se ramènent au problème de programmation linéaire suivant.

Minimiser la forme linéaire

$$L(X)=\sum_{j=1}^{n} c_j x_j+\sum_{j=1}^{n}\sum_{i=1}^{l_j} \sigma_{ij} x_j^{(i)} \tag{1.33}$$

sous les conditions

$$\sum_{j=1}^{n} a_{ij} x_j = b_i, \quad i=1, 2, \ldots, m, \tag{1.34}$$

$$\alpha_j \leqslant x_j \leqslant \beta_j, \quad j = 1, 2, \ldots, n, \tag{1.35}$$

$$\left.\begin{array}{l} x_j \leqslant d_{ij}+x_j^{(i)} \\ x_j^{(i)} \geqslant 0 \end{array}\right\} \quad i=1, 2, \ldots, l_j; \quad j=1, 2, \ldots, n. \tag{1.36}$$

Le problème (1.33)-(1.36) peut avoir, par exemple, l'interprétation économique suivante.

Soient x_j le volume de production de la j-ième branche; c_j, le coût unitaire de production de la j-ième branche.

Les conditions (1.34), qui associent les variables x_j du problème, sont des relations de bilan entre la production des différentes branches. Les contraintes (1.34) garantissent le volume de production demandé aux branches qui fournissent le produit fini. Les conditions (1.35) établissent les contraintes sur les volumes de production de chaque branche, contraintes engendrées, par exemple, par l'étude de la demande. Les conditions (1.36) traduisent pour chaque branche (j) la liaison entre le volume de production x_j et la demande supplémentaire $x_j^{(i)}$ en ressources de différents types (i). Les ressources ainsi limitées varient d'une branche à l'autre.

Les ressources supplémentaires (du i-ème type) sont nécessaires pour que le volume de production x_j puisse dépasser un certain niveau fixé d_{ij}. Les quantités $\sigma_{ij} > 0$ définissent les frais supplémentaires qui, dans la j-ième branche, sont liées à l'utilisation d'une unité de la i-ème ressource en sus de la norme prévue. Le facteur de qualité de la planification (1.33), à minimiser, caractérise le coût total de production.

Voici les traits particuliers caractéristiques du problème (1.33)-(1.36). Toutes ses conditions peuvent être rangées en deux groupes. Celles du premier groupe forment un système arbitraire d'égalités

ou d'inégalités linéaires, et celles du deuxième sont de la forme

$$\alpha_j \leqslant x_j \leqslant \beta_j;$$
$$x_j \leqslant d_{ij} + x_j^{(i)},$$
$$x_j^{(i)} \geqslant 0.$$

Les conditions du premier groupe ne comptent que des variables x_j. Celles du deuxième contiennent en outre les variables $x_j^{(i)}$.

Nous avons déjà dit que les quantités σ_{ij} sont supposées positives.

Il est logique de considérer que les nombres d_{ij} sont tels que

$$\alpha_j < d_{1j} < d_{2j} < \ldots < d_{l_j j} < \beta_j.$$

Dans la plupart des cas pratiques, les bornes inférieures α_j des conditions (1.35) sont égales à zéro. Les bornes supérieures, dans certaines des conditions (1.35), sont quelquefois absentes. Dans de tels cas, les β_j correspondantes sont posées égales à l'infini. Pour condenser l'exposé, nous désignerons par la suite le problème (1.33)-(1.36) par la lettre (A). Nous nommerons, comme d'habitude, programmes du problème (A) les collections de nombres $(x_j, x_j^{(i)})$ qui vérifient les conditions (1.34)-(1.36). Nous dirons que le programme du problème (A) qui minimise la forme linéaire (1.33) est le programme optimal ou la solution du problème.

Nous avons déjà dit que le problème (A) se distingue par ses conditions linéaires du type général (1.34)-(1.35) qui n'associent entre elles qu'une partie de variables du problème (variables $x_1, x_2, \ldots, x_n$). Les autres variables du problème (leur majorité) sont assujetties aux conditions simples (1.36). Aussi ne serait-il pas rationnel, lors de la résolution du problème (A), d'utiliser les méthodes de programmation linéaire sans tenir compte de cette particularité absolument essentielle du problème. L'idée maîtresse de la méthode exposée dans ce qui suit consiste à remplacer le problème (A) par un problème non linéaire de minimisation aux variables $x_1, x_2, \ldots, x_n$ vérifiées par les conditions (1.34), (1.35).

Ainsi, au prix d'une certaine complication de la fonction à minimiser, nous nous débarrassons des variables $x_j^{(i)}$ et des conditions (1.36). Le problème non linéaire obtenu peut être résolu à l'aide d'une méthode assez économique, dont la description est donnée dans les paragraphes suivants (§§ 3 à 6). Il ne reste ensuite qu'à reprendre le problème étudié (A) et à écrire la solution cherchée en se servant de formules simples.

Enonçons le problème non linéaire qui, comme on le montrera, est équivalent au problème initial (A).

Calculer le vecteur $X = (x_1, x_2, \ldots, x_n)$ minimisant la fonction

$$F(X) = \sum_{j=1}^{n} f_j(x_j) \tag{1.37}$$

sous les conditions (1.34) et (1.35).

Les fonctions $f_j(x_j)$ sont définies par la formule

$$f_j(x_j) = \begin{cases} c_j x_j, & \alpha_j \leqslant x_j \leqslant d_{1j}, \\ (c_j + \sigma_{1j})(x_j - d_{1j}) + f_j(d_{1j}), & d_{1j} \leqslant x_j \leqslant d_{2j}, \\ \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots & \\ \left(c_j + \sum_{i=1}^{l_j} \sigma_{ij}\right)(x_j - d_{l_j j}) + f_j(d_{l_j j}), & d_{l_j j} \leqslant x_j \leqslant \beta_j. \end{cases} \tag{1.38}$$

Posons

$$c_{ij} = c_j + \sum_{\alpha=1}^{i} \sigma_{\alpha_j}, \quad i = 1, 2, \ldots, l_j,$$

$$c_{0j} = c_j.$$

Ainsi, quel que soit j, la fonction $f_j(x_j)$ est continue sur $[\alpha_j, \beta_j]$ et linéaire sur chacun des segments $[d_{kj}, d_{k+1,j}]$; de plus, sur $[d_{kj}, d_{k+1,j}]$ son coefficient angulaire est égal à c_{kj}. Notons que comme, par condition, tout $\sigma_{kj} > 0$, les $f_j(x_j)$ sont des fonctions convexes vers le bas. Désignons par (B) le problème de minimisation de la fonction (1.37) sous les conditions (1.34), (1.35). Dans les notations du paragraphe 1.1, le problème (B) coïncide avec le problème I. Appliquant au problème (B) la terminologie de la programmation linéaire, appelons le vecteur $X = (x_1, x_2, \ldots, x_n)$ satisfaisant aux conditions (1.34), (1.35) programme du problème (B), et le programme X minimisant la fonction $F(X)$, programme optimal ou solution du problème.

Etablissons l'équivalence des problèmes (A) et (B), dont voici le sens. Si $\overline{X} = (\overline{x}_1, \overline{x}_2, \ldots, \overline{x}_n)$ est le programme optimal du problème (B) et si $d_{k_j j} \leqslant \overline{x}_j \leqslant d_{k_j+1,j}$, la collection des variables $(\overline{x}_j, \overline{x}_j^{(i)})$, où

$$\overline{x}_j^{(i)} = \begin{cases} \overline{x}_j - d_{ij}, & i \leqslant k_j, \\ 0, & i > k_j, \end{cases} \tag{1.39}$$

donne la solution du problème (A).

Inversement, si $(\overline{x}_j, \overline{x}_j^{(i)})$ est la solution du problème (A), le vecteur $\overline{X} = (\overline{x}_1, \ldots, \overline{x}_n)$ est le programme optimal du problème (B). Constatons avant tout qu'en raison de la positivité des nombres σ_{ij}, la solution du problème (A) doit être recherchée parmi ceux

de ses programmes $(x_j, x_j^{(i)})$ qui satisfont aux conditions

$$x_j = d_{ij} + x_j^{(i)} \quad \text{si} \quad x_j^{(i)} > 0. \tag{1.40}$$

Si $(x_j, x_j^{(i)})$ vérifie la condition (1.40),

$$\sum_{j=1}^{n} c_j x_j + \sum_{j=1}^{n} \sum_{i=1}^{l_j} \sigma_{ij} x_j^{(i)} = \sum_{j=1}^{n} [c_j x_j + \sum_{i=1}^{k_j} \sigma_{ij} (x_j - d_{ij})],$$

où k_j est égal au plus grand des indices i tels que $d_{ij} < x_j$. La définition de la fonction $f_j(x_j)$ entraîne

$$\sum_{j=1}^{n} c_j x_j + \sum_{j=1}^{n} \sum_{i=1}^{l_j} \sigma_{ij} x_j^{(i)} = F(X) = \sum_{j=1}^{n} f_j(x_j). \tag{1.41}$$

Soit maintenant $\overline{X} = (\overline{x}_1, \ldots, \overline{x}_n)$ la solution du problème (B). Déterminons $\overline{x}_j^{(i)}$ d'après la formule (1.39). Par construction, le programme $(\overline{x}_j, \overline{x}_j^{(i)})$ satisfait à la condition (1.40). Ceci dit, si $(x_j, x_j^{(i)})$ est un programme arbitraire du problème (A) qui justifie la condition (1.40), utilisons l'inégalité (1.41), pour obtenir

$$\sum_{j=1}^{n} c_j x_j + \sum_{j=1}^{n} \sum_{i=1}^{l_j} \sigma_{ij} x_j^{(i)} = F(X) \geqslant F(\overline{X}) = \sum_{j=1}^{n} c_j \overline{x}_j + \sum_{j=1}^{n} \sum_{i=1}^{l_j} \sigma_{ij} \overline{x}_j^{(i)},$$

ce qui signifie que le programme $(\overline{x}_j, \overline{x}_j^{(i)})$ est optimal. Inversement, soit $(\overline{x}_j, \overline{x}_j^{(i)})$ le programme optimal du problème (A). Ce programme satisfaisant nécessairement à la condition (1.40), il vérifie l'égalité (1.41). Considérons un vecteur arbitraire $X = (x_1, x_2, \ldots, x_n)$ qui vérifie les contraintes (1.34), (1.35). Si l'on détermine à l'aide de ce vecteur les nombres $x_j^{(i)}$ suivant la formule (1.39), on obtient le programme $(x_j, x_j^{(i)})$ du problème (A) qui vérifie la condition (1.40) et, par conséquent, l'égalité (1.41). On en tire en utilisant l'optimalité du programme $(\overline{x}_j, \overline{x}_j^{(i)})$:

$$F(X) \geqslant F(\overline{X}).$$

1.4. Le problème (1.33)-(1.36) correspond au modèle de planification du volume de production d'un groupe de branches associées de l'économie. Il se peut que l'augmentation de la fabrication de produit par ces branches soit limitée par des ressources en provenance de l'extérieur. Supposons qu'il y ait s types de ces ressources. Le volume de la k-ième ressource qui peut être utilisé dans le groupe de branches concernées ne dépasse pas g_k $(k = 1, 2, \ldots, s)$ unités.

Posons pour simplifier le modèle, que le coût associé à l'utilisation des ressources en provenance de l'extérieur est pris en compte lors de la planification à un niveau plus élevé et ne doit pas être

considéré en recherchant le facteur de qualité de la solution du problème.

Les différentes restrictions imposées au niveau de production de la j-ième branche définissent le coût différent de la k-ième ressource par unité de produit. On peut indiquer des volumes critiques de production $d_{kj}^{(t)}$ $(t = 1, 2, \ldots, r_{kj})$, tels que pour

$$d_{kj}^{(t)} \leqslant x_j \leqslant d_{kj}^{(t+1)}$$

le coût de la k-ième ressource par unité de produit ne change pas, alors qu'il augmente pour $x_j > d_{kj}^{(t+1)}$.

Supposons que pour $\alpha_j \leqslant x_j \leqslant d_{kj}^{(1)}$, p_{kj} constitue le coût de la k-ième ressource par unité de produit fourni par la j-ième branche, et que, dans le cas d'un volume de production x_j dépassant le niveau $d_{kj}^{(t)}$ $(t = 1, 2, \ldots, r_{kj})$, le coût de la k-ième ressource par unité du j-ième produit augmente de $\gamma_{kj}^{(t)}$.

Il est naturel de considérer que

$$\alpha_j < d_{kj}^{(1)} < d_{kj}^{2} < \ldots < d_{kj}^{(r_{kj})} < \beta_j.$$

Pour tenir compte, dans le modèle (1.33)-(1.36), des contraintes sur les ressources en provenance de l'extérieur, il faut ajouter aux conditions (1.34)-(1.36) les systèmes de contraintes suivants:

$$\sum_{j=1}^{n} p_{kj}x_j + \sum_{j=1}^{n} \sum_{t=1}^{r_{kj}} \gamma_{kj}^{(t)} x_{jk}^{(t)} \leqslant g_k, \quad k = 1, 2, \ldots, s, \tag{1.42}$$

$$\left.\begin{array}{l} x_j \leqslant x_{jk}^{(t)} + d_{jk}^{(t)} \\ x_{jk}^{(t)} \geqslant 0 \end{array}\right\} \begin{array}{l} t = 1, 2, \ldots, r_{kj}, \\ k = 1, 2, \ldots, s;\ j = 1, 2, \ldots, n. \end{array} \tag{1.43}$$

$x_{jk}^{(t)}$ est ici l'excédent du volume de production de la j-ième branche sur le niveau critique $d_{kj}^{(t)}$ lié à la mise en œuvre de la k-ième ressource extérieure.

Désignons par (C) le problème (1.33)-(1.36), (1.42)-(1.43).

Donnons l'énoncé du problème linéaire par morceaux [nous l'appellerons problème (D)] équivalent au problème (C).

Calculer le vecteur $X = (x_1, x_2, \ldots, x_n)$ minimisant la fonction (1.37) sous les conditions (1.34)-(1.35) et le système supplémentaire de contraintes du type

$$\sum_{j=1}^{n} g_{kj}(x_j) \leqslant g_k, \quad k = 1, 2, \ldots, s, \tag{1.44}$$

où les fonctions linéaires par morceaux $g_{kj}(x_j)$ sont définies par les formules du type

$$g_{kj}(x_j)=\begin{cases} p_{kj}x_j & \text{pour } \alpha_j \leqslant x_j \leqslant d_{kj}^{(1)}, \\ (p_{kj}+\gamma_{kj}^{(1)})(x_j-d_{kj}^{(1)})+g_{kj}(d_{kj}^{(1)}) & \text{pour } d_{kj}^{(1)} \leqslant x_j \leqslant d_{kj}^{(2)}, \\ \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots & \\ \left(p_{kj}+\sum\limits_{t=1}^{r_{kj}} \gamma_{kj}^{(t)}\right)(x_j-d_{kj}^{(r_{kj})})+g_{kj}(d_{kj}^{(r_{kj})}) & \text{pour } d_{kj}^{(r_{kj})} \leqslant x_j \leqslant \beta_j. \end{cases} \tag{1.45}$$

Dans les notations du paragraphe 1.1, le problème (D) coïncide avec le problème III. L'équivalence des problèmes (C) et (D) s'établit sans peine. Il suffit pour cela de retenir, en plus des considérations du paragraphe précédent, le fait que les fonctions $g_{kj}(x_j)$ sont, par définition (1.45), convexes vers le bas. Nous laissons au lecteur le soin de construire toutes les démonstrations exigées (cf. exercice 2).

Remarquons qu'il se trouve des travaux ([2], [58], par exemple) dans lesquels les problèmes linéaires par morceaux sont ramenés à des problèmes linéaires. Toutefois, dans tous ces cas, le nombre de variables et de contraintes des problèmes linéaires dépasse considérablement le nombre de variables et de conditions des problèmes linéaires par morceaux équivalents. Si l'on a à sa disposition des méthodes et des programmes de résolution des problèmes linéaires par morceaux, le passage aux problèmes équivalents de programmation linéaire n'est guère opportun.

§ 2. Critères d'optimalité des problèmes de programmation linéaire par morceaux

2.1. Considérons le problème général de programmation linéaire par morceaux. Calculer le vecteur $X = (x_1, x_2, \dots, x_n)$, de dimension n, minimisant la fonction

$$F(X) \tag{2.1}$$

sous les conditions

$$G(X) \leqslant 0, \tag{2.2}$$

$$AX - B = 0, \tag{2.3}$$

$$X \in R, \tag{2.4}$$

où $F(X)$ et les composantes $g_i(X)$ $(i = 1, 2, \dots, m_1)$ de la fonction vectorielle $G(X)$ sont les fonctions linéaires par morceaux des variables $x_1, x_2, \dots, x_n$;

A, la matrice de dimension $(m - m_1) \times n$;

B, le vecteur de dimension $(m - m_1)$;

R, le polyèdre convexe.

Le vecteur X satisfaisant aux conditions (2.2)-(2.4) est dit programme du problème linéaire par morceaux. Nous appellerons, comme d'habitude, programme optimal ou solution du problème le programme qui rend minimale la fonction (2.1).

Le théorème suivant définit le critère d'optimalité du problème général de programmation linéaire par morceaux.

Théorème 2.1. *Le programme X^* du problème* (2.1)-(2.4) *en est une solution si et seulement si il existe un vecteur de dimension m $\Lambda = (\Lambda_1, \Lambda_2)$, $\Lambda_1 = (\lambda_1, \ldots, \lambda_{m_1}) \geqslant 0$, $\Lambda_2 = (\lambda_{m_1+1}, \ldots, \lambda_m)$, tel que*

a) la fonction $\varphi(X, \Lambda) = F(X) + (\Lambda_1, G(X)) + (\Lambda_2 (AX - B))$ (2.5)

atteigne en X^* le minimum par rapport à X sur l'ensemble R et

$$b) \quad (\Lambda_1, G(X^*)) = 0. \tag{2.6}$$

Démonstration. Démontrons que les conditions du théorème sont suffisantes. Supposons qu'il existe des vecteurs $\Lambda_1 \geqslant 0$ et Λ_2 qui vérifient les conditions du théorème. En vertu de a),

$$\varphi(X^*, \Lambda) \leqslant \varphi(X, \Lambda) \quad \text{pour tout} \quad X \in R,$$

ou, ce qui revient au même,

$$F(X^*) + (\Lambda_1, G(X^*)) + (\Lambda_2, (AX^* - B)) \leqslant$$
$$\leqslant F(X) + (\Lambda_1, G(X)) + (\Lambda_2, (AX - B)) \tag{2.7}$$

pour tout $X \in R$.

Le vecteur X^* est un programme du problème (2.1)-(2.4). Par conséquent $AX^* - B = 0$. En vertu de b),

$$(\Lambda_1, G(X^*)) = 0.$$

Pour un programme X quelconque du problème (2.1)-(2.4),

$$AX - B = 0 \quad \text{et} \quad (\Lambda_1, G(X)) \leqslant 0.$$

Dans ces conditions, l'inégalité (2.7) amène

$$F(X^*) \leqslant F(X)$$

pour tout programme X du problème (2.1)-(2.4). Les conditions du théorème sont ainsi suffisantes.

Prouvons que ces conditions sont nécessaires.

1. Considérons dans un espace de dimension $(m + 1)$ un ensemble M des points $Z = (z_0, z_1, \ldots, z_m)$, défini de la façon suivante. Le point $Z \in M$ s'il se trouve un vecteur $X \in R$ tel que

$$z_0 \geqslant F(X), \tag{2.8}$$

$$z_i \geqslant g_i(X), \quad i = 1, 2, \ldots, m_1, \tag{2.9}$$

$$z_i = (AX - B)_i = \sum_{j=1}^{n} a_{ij}x_j - b_i, \; i = m_1 + 1, \ldots, m. \tag{2.10}$$

L'ensemble M est un polyèdre convexe. Ceci résulte des raisonnements que voici. Considérons dans l'espace de dimension $(m + n + 1)$ l'ensemble $\widetilde{M}$ des points $\widetilde{Z} = (Z, X)$ dont les coordonnées vérifient les relations (2.8)-(2.10) et, de plus, $X \in R$. Les fonctions linéaires par morceaux convexes vers le bas $F(X)$ et $g_i(X)$ peuvent s'écrire, par définition, sous la forme (1.1). Par suite, on peut présenter les inégalités (2.8), (2.9) sous la forme d'un système d'inégalités linéaires. L'ensemble $\widetilde{M}$ des points $\widetilde{Z}$ dont les coordonnées vérifient ce système d'égalités et d'inégalités linéaires est un polyèdre convexe. L'ensemble M, en tant que projection du polyèdre convexe $\widetilde{M}$ sur un espace de dimension $(m + 1)$, est également un polyèdre convexe.

Pour le voir il suffit, après s'être servi de la deuxième définition du polyèdre convexe (cf. [87], chapitre 5, § 2), de représenter $\widetilde{M}$ comme l'ensemble des points $\widetilde{Z}$ du type

$$\widetilde{Z} = \sum_{k=1}^{N_1} \widetilde{Z}_k \alpha_k + \sum_{k=1}^{N_2} \widetilde{Z}_{N_1+k} \beta_k, \quad \alpha_k \geqslant 0, \quad \beta_k \geqslant 0,$$

$$\sum_{k=1}^{N_1} \alpha_k = 1.$$

Si dans la représentation donnée pour les points $\widetilde{Z}$ de l'espace de dimension $(m + n + 1)$ on considère les $m + 1$ premières coordonnées, il vient

$$Z = \sum_{k=1}^{N_1} Z_k \alpha_k + \sum_{k=1}^{N_2} Z_{N_1+k} \beta_k, \quad \alpha_k \geqslant 0, \quad \beta_k \geqslant 0,$$

$$\sum_{k=1}^{N_1} \alpha_k = 1,$$

où le premier membre est un point quelconque $Z \in M$.

Par conséquent, M est un polyèdre convexe.

2. Définissons dans l'espace de dimension $(m + 1)$ la demi-droite L:

$$z_0 \leqslant F(X^*), \tag{2.11}$$

$$z_i = 0, \quad i = 1, 2, \ldots, m, \tag{2.12}$$

où X^* est le programme optimal du problème (2.1)-(2.4).

La demi-droite L et l'ensemble M ont un point commun unique. C'est le point

$$Z^* = \{F(X^*), \underbrace{0, \ldots, 0}_{m}\},$$

extrémité de la demi-droite L.

En effet, supposons qu'il n'en soit pas ainsi et qu'il y ait un point Z' différent de Z^* qui appartienne à la demi-droite L et à l'ensemble M.

Par définition de la demi-droite L et de l'ensemble M, l'hypothèse adoptée signifie qu'il existe un point $X \in R$ tel que

$$F(X^*) > z_0' \geqslant F(X),$$
$$0 \geqslant g_i(X), \quad i = 1, 2, \ldots, m_1,$$
$$0 = (AX - B)_i, \quad i = m_1 + 1, \ldots, m.$$

Les relations obtenues contredisent le fait que le vecteur X^* est solution du problème (2.1)-(2.4).

3. Le polyèdre convexe M est l'intersection d'un nombre fini de demi-espaces et d'hyperplans. Mettons à part, parmi les hyperplans frontières de l'ensemble M, les hyperplans contenant le point Z^*, extrémité de la demi-droite L. Si tous ces hyperplans contenaient, outre le point Z^*, toute la demi-droite L, au moins un segment de la demi-droite L appartiendrait, en plus du point Z^*, à l'ensemble M. Mais c'est impossible du fait que d'après ce qu'on a démontré Z^* est l'unique point commun à la demi-droite L et à l'ensemble M. Par conséquent, parmi les hyperplans retirés il existe un hyperplan Π qui passe par le point Z^*, sans contenir la demi-droite L. L'hyperplan Π étant frontière, il est hyperplan d'appui pour M en Z^*.

Soit

$$\sum_{i=0}^{m} r_i z_i = \text{const}$$

l'équation de l'hyperplan Π.

Le point $Z^* \in \Pi$. Aussi l'équation de cet hyperplan peut-elle être récrite sous la forme

$$\sum_{i=0}^{m} r_i z_i = r_0 z_0^*,$$

soit

$$\sum_{i=0}^{m} r_i z_i = r_0 F(X^*). \tag{2.13}$$

Considérons que les coefficients r_i qui définissent Π sont choisis de telle façon que pour tout point $Z \in M$

$$\sum_{i=0}^{m} r_i z_i \geqslant r_0 F(X^*). \tag{2.14}$$

On voit aisément que

$$r_i \geqslant 0 \quad \text{avec} \quad i = 0, 1, \ldots, m_1. \tag{2.15}$$

En effet, si l'un de ces nombres était négatif, alors, pour des valeurs suffisamment grandes des coordonnées correspondantes z_i du point $Z \in M$, le premier membre de l'inégalité (2.14) deviendrait un nombre aussi petit que l'on veut, ce qui contredit (2.14).

L'hyperplan Π ne contient pas la demi-droite L. Par suite, $r_0 \neq 0$, et, en vertu de (2.15),

$$r_0 > 0.$$

Introduisons les notations

$$\lambda_i = \frac{r_i}{r_0}, \quad (i = 1, 2, \ldots, m)$$

$$\Lambda_1 = (\lambda_1, \ldots, \lambda_{m_1}); \quad \Lambda_2 = (\lambda_{m_1+1}, \ldots, \lambda_m).$$

Les inégalités (2.15) entraînent

$$\Lambda_1 \geqslant 0. \tag{2.16}$$

Pour les points $Z \in M$ définis par les égalités

$$z_0 = F(X),$$
$$z_i = g_i(X), \quad i = 1, 2, \ldots, m_1,$$
$$z_i = (AX \quad B)_i, \quad i = m_1 + 1, \ldots, m,$$

où $X \in R$, l'inégalité (2.14) se met sous la forme

$$F(X) + (\Lambda_1, G(X)) + (\Lambda_2, (AX - B)) \geqslant F(X^*), \tag{2.17}$$
$$X \in R.$$

Le vecteur X^* est le programme optimal du problème (2.1)-(2.4). En raison de (2.2), (2.3) et (2.16), l'inégalité (2.17) conduit à

$$F(X) + (\Lambda_1, G(X)) + (\Lambda_2, (AX - B)) \geqslant$$
$$\geqslant F(X^*) + (\Lambda_1, G(X^*)) + (\Lambda_2, (AX^* - B)), \quad X \in R, \tag{2.18}$$

ou, ce qui revient au même par suite de (2.5),

$$\varphi(X^*, \Lambda) \leqslant \varphi(X, \Lambda), \quad X \in R. \tag{2.19}$$

On a ainsi démontré que si le vecteur X^* est solution du problème (2.1)-(2.4), il existe un vecteur $\Lambda = (\Lambda_1, \Lambda_2)$, $\Lambda_1 \geqslant 0$, tel que ce même vecteur X^* assure le minimum de la fonction $\varphi(X, \Lambda)$ sur l'ensemble R. Il est ainsi établi que la condition a) du théorème est nécessaire.

Pour s'assurer que la condition b) est nécessaire elle aussi, posons dans le premier membre de l'inégalité (2.17) $X = X^*$. Il vient

$$(\Lambda_1, G(X^*)) \geqslant 0. \tag{2.20}$$

Mais $\Lambda_1 \geqslant 0$, et $G(X^*) \leqslant 0$. Ceci signifie que l'inégalité (2.20) ne peut être vérifiée qu'avec

$$(\Lambda_1, G(X^*)) = 0.$$

Ce qu'il fallait démontrer.

2.2. Le théorème 2.1. peut servir de base pour formuler des critères d'optimalité des programmes des problèmes de programmation linéaire par morceaux, commodes pour les schémas de calcul.

Considérons les différentes formulations particulières de problèmes de programmation linéaire par morceaux fréquentes dans les applications. Commençons par le problème I (par le problème (B), dans les termes du paragraphe 1.3), dans lequel le facteur de qualité est de la forme

$$F(X) = \sum_{j=1}^{n} f_j(x_j), \tag{2.21}$$

où $f_j(x_j)$ sont des fonctions linéaires par morceaux, convexes vers le bas de leurs arguments, alors que les contraintes constituent le système de conditions linéaires

$$AX - B = 0, \tag{2.22}$$

$$\alpha_j \leqslant x_j \leqslant \beta_j. \tag{2.23}$$

D'après le théorème 2.1, le programme X^* du problème (2.21)-(2.23) en est une solution si et seulement si l'on trouve un vecteur Λ tel que la fonction

$$\varphi(X, \Lambda) = F(X) + (\Lambda, (AX - B))$$

soit minimisée en X^* sur l'ensemble (2.23). Autrement dit, la condition nécessaire et suffisante pour que le programme X^* soit optimal consiste en l'existence d'un vecteur Λ vérifiant les conditions

$$F(X^*) + (\Lambda, (AX^* - B)) \leqslant F(X) + (\Lambda, (AX - B)) \tag{2.24}$$

$$\text{avec} \quad \alpha_j \leqslant x_j \leqslant \beta_j.$$

Récrivons l'inégalité (2.24) sous la forme

$$\sum_{j=1}^{n} f_j(x_j^*) + \sum_{j=1}^{n} (\Lambda, A_j) x_j^* \leqslant \sum_{j=1}^{n} f_j(x_j) + \sum_{j=1}^{n} (\Lambda, A_j) x_j \tag{2.25}$$

$$\text{avec} \quad \alpha_j \leqslant x_j \leqslant \beta_j.$$

Soit $x_k^* < \beta_k$. Posons $x_k = x_k^* + \varepsilon$ $(\varepsilon > 0)$, et $x_j = x_j^*$ avec $j \neq k$. L'inégalité (2.25) s'écrit alors

$$-(\Lambda, A_k)\varepsilon \leqslant f_k(x_k^* + \varepsilon) - f_k(x_k^*).$$

On en tire

$$f_k^{(+)}(x_k^*) \geqslant -(\Lambda, A_k), \tag{2.26}$$

où $f_k^{(+)}(x_k^*)$ est la dérivée à droite de la fonction f_k au point x_k^*.

Soit maintenant $x_k^* > \alpha_k$. En posant $x_k = x_k^* - \varepsilon$ $(\varepsilon > 0)$ et $x_j = x_j^*$ avec $j \neq k$, on obtient d'une façon analogue

$$f_k^{(-)}(x_k^*) \leqslant -(\Lambda, A_k), \tag{2.27}$$

où $f_k^{(-)}(x_k^*)$ est la dérivée à gauche de la fonction f_k en x_k^*.

Le signe de Λ n'étant pas spécifié, on peut omettre dans les inégalités (2.26) et (2.27) le signe moins devant Λ.

Ainsi, on tire de l'inégalité (2.25):

pour $\alpha_j < x_j^* < \beta_j$,

$$f_j^{(-)}(x_j^*) \leqslant (\Lambda, A_j) \leqslant f_j^{(+)}(x_j^*); \tag{2.28}$$

pour $x_j^* = \alpha_j$,

$$f_j^{(+)}(x_j^*) \geqslant (\Lambda, A_j); \tag{2.29}$$

pour $x_j^* = \beta_j$,

$$f_j^{(-)}(x_j^*) \leqslant (\Lambda, A_j). \tag{2.30}$$

On voit aisément que des inégalités (2.28)-(2.30) résulte l'inégalité (2.25) (cf. exercice 3).

Donc, pour qu'un programme X^* du problème (2.21)-(2.23) en soit une solution, il faut et il suffit qu'il existe un vecteur Λ de dimension m satisfaisant aux relations (2.28)-(2.30).

2.3. Considérons maintenant le problème linéaire par morceaux II. Appliqué au problème II, le théorème 2.1 s'énonce de la façon suivante.

Un programme $X^* = (x_1^*, \ldots, x_n^*)$ du problème II est optimal si et seulement si il existe un vecteur $\Lambda = (\lambda_1, \ldots, \lambda_m) \geqslant 0$ qui vérifie l'inégalité

$$\sum_{j=1}^{n} f_j(x_j^*) + \sum_{i=1}^{m}\sum_{j=1}^{n} \lambda_i g_{ij}(x_j^*) \leqslant \sum_{j=1}^{n} f_j(x_j) + \sum_{i=1}^{m}\sum_{j=1}^{n} \lambda_i g_{ij}(x_j) \tag{2.31}$$

pour tout $\alpha_j \leqslant x_j \leqslant \beta_j$, $j = 1, 2, \ldots, n$, et

$$\sum_{i=1}^{m}\sum_{j=1}^{n} \lambda_i g_{ij}(x_{ij}^*) = 0. \tag{2.32}$$

En posant, comme dans le paragraphe précédent, $x_k = x_k^* + \varepsilon$ $(\varepsilon > 0)$ avec $x_k < \beta_k$ et $x_j = x_j^*$ $(j \neq k)$, puis $x_k = x_k^* - \varepsilon$ $(\varepsilon > 0)$ avec $x_k > \alpha_k$ et $x_j = x_j^*$ $(j \neq k)$, on obtient à partir de l'inégalité (2.31):

avec $\alpha_j < x_j < \beta_j$,

$$f_j^{(+)}(x_j^*) + (\Lambda, g_j^{(+)}(x_j^*)) \geqslant 0,$$

$$f_j^{(-)}(x_j^*) + (\Lambda, g_j^{(-)}(x_j^*)) \leqslant 0,$$

ou, ce qui revient au même,

$$f_j^{(-)}(x_j^*) + (\Lambda, g_j^{(-)}(x_j^*)) \leqslant 0 \leqslant f_j^{(+)}(x_j^*) + (\Lambda, g_j^{(+)}(x_j^*)); \tag{2.33}$$

avec $x_j^* = \alpha_j$,

$$f_j^{(+)}(x_j^*) + (\Lambda, g_j^{(+)}(x_j^*)) \geqslant 0, \tag{2.34}$$

et avec $x_j^* = \beta_j$,

$$f_j^{(-)}(x_j^*) + (\Lambda, g_j^{(-)}(x_j^*)) \leqslant 0. \tag{2.35}$$

Ici

$$g_j^{(+)}(x_j^*) = \{g_{1j}^{(+)}(x_j^*), \ldots, g_{mj}^{(+)}(x_j^*)\}$$

et

$$g_j^{(-)}(x_j^*) = \{g_{1j}^{(-)}(x_j^*), \ldots, g_{mj}^{(-)}(x_j^*)\}$$

sont les vecteurs de dimension m, dont les composantes sont respectivement les dérivées à droite et à gauche des fonctions linéaires par morceaux $g_{ij}(x_j)$ en x_j^*.

Les inégalités (2.33)-(2.35) entraînent l'inégalité (2.31) (cf. exercice 4).

Ainsi avec $\Lambda \geqslant 0$, les inégalités (2.33)-(2.35) constituent avec l'égalité (2.32) les conditions d'optimalité nécessaires et suffisantes d'un programme du problème II. Le critère d'optimalité obtenu du programme est à la base du calcul numérique du problème linéaire par morceaux considéré.

Le problème III (dans les termes du paragraphe 1.3, le problème (D)) ne diffère du problème II que par le système supplémentaire de conditions-égalités linéaires. Des raisonnements analogues aux précédents conduisent au critère d'optimalité suivant d'un programme du problème III.

Le programme $X^* = (x_1^*, \ldots, x_n^*)$ du problème III en est une solution si et seulement si il existe un vecteur de dimension m_1, $\Lambda_1 = (\lambda_1, \ldots, \lambda_{m_1}) \geqslant 0$, et un vecteur de dimension $(m - m_1)$, $\Lambda_2 = (\lambda_{m_1+1}, \ldots, \lambda_m)$, tels que

$$f_j^{(-)}(x_j^*) + (\Lambda_1, g_j^{(-)}(x_j^*)) \leqslant (\Lambda_2, A_j) \leqslant$$
$$\leqslant f_j^{(+)}(x_j^*) + (\Lambda_1, g_j^{(+)}(x_j^*)) \tag{2.36}$$
$$\text{avec} \quad \alpha_j < x_j^* < \beta_j,$$

$$f_j^{(+)}(x_j^*) + (\Lambda_1, g_j^{(+)}(x_j^*)) \geqslant (\Lambda_2, A_j) \tag{2.37}$$
$$\text{avec} \quad x_j^* = \alpha_j,$$

$$f_j^{(-)}(x_j^*) + (\Lambda_1, g_j^{(-)}(x_j^*)) \leqslant (\Lambda_2, A_j) \tag{2.38}$$
$$\text{avec} \quad x_j^* = \beta_j,$$

$$\sum_{i=1}^{m_1} \sum_{j=1}^{n} \lambda_i^{(1)} g_{ij}(x_j^*) = 0. \tag{2.39}$$

Nous avons conservé ici les mêmes notations que dans les problèmes précédents.

2.4. Enonçons le critère d'optimalité du problème IV. Dans ce problème, la fonction économique et une partie des contraintes apparaissent comme une somme de fonctions linéaires par morceaux de formes linéaires. Certains cas particuliers de problèmes de ce type, dans lesquels les fonctions linéaires par morceaux sont les modules de formes linéaires, ont été explorés dans [32] et [42].

En remplaçant les variables, on peut ramener le problème IV au problème du type III.

Introduisons les variables

$$y_{ik} = L_{ik}(X) = \sum_{j=1}^{n} a_{ij}^{(k)} x_j - b_i^{(k)}, \tag{2.40}$$

$$k = 1, 2, \ldots, s_i; \quad i = 0, 1, 2, \ldots, m_1.$$

Le problème IV s'écrit alors

$$\sum_{k=1}^{s_0} f_k(y_{0k}) \to \min, \tag{2.41}$$

$$\sum_{k=1}^{s_i} g_{ik}(y_{ik}) \leqslant 0; \quad i = 1, 2, \ldots, m_1, \tag{2.42}$$

$$\sum_{j=1}^{n} a_{ij}^{(k)} x_j - y_{ik} = b_i^{(k)}; \quad k = 1, 2, \ldots, s_i; \quad i = 0, 1, 2, \ldots, m_1, \tag{2.43}$$

$$\sum_{j=1}^{n} a_{ij} x_j - b_i = 0; \quad i = m_1 + 1, \ldots, m, \tag{2.44}$$

$$\alpha_j \leqslant x_j \leqslant \beta_j; \quad j = 1, 2, \ldots, n. \tag{2.45}$$

Soit

$$Z = \{y_{01}, \ldots, y_{0s_0}, y_{11}, \ldots, y_{1s_1}, \ldots, y_{m_1 1}, \ldots, y_{m_1 s_{m_1}}, x_1, \ldots, x_n\} \tag{2.46}$$

le programme arbitraire du problème (2.41)-(2.45).

D'après le critère d'optimalité obtenu dans ce qui précède pour le problème III, le programme

$$Z^* = \{y_{01}^*, \ldots, y_{0s_0}^*, y_{11}^*, \ldots, y_{1s_1}^*, \ldots, y_{m_1 1}^*, \ldots, y_{m_1 s_{m_1}}^*, x_1^*, \ldots, x_n^*\}$$

du problème (2.41)-(2.45) est son programme optimal si et seulement si il est possible de composer des collections de nombres

$$\lambda_i \geqslant 0 \quad (i = 1, 2, \ldots, m_1); \quad \lambda_{ik} \ (k = 1, 2, \ldots, s_i;$$
$$i = 0, 1, \ldots, m_1),$$
$$\lambda_i \ (i = m_1 + 1, \ldots, m),$$

telles qu'elles vérifient les relations suivantes:

$$f_k^{(-)}(y_{0k}^*) \leqslant \lambda_{0k} \leqslant f_k^{(+)}(y_{0k}^*), \quad k = 1, 2, \ldots, s_0, \tag{2.47}$$

$$\lambda_i g_{ik}^{(-)}(y_{ik}^*) \leqslant \lambda_{ik} \leqslant \lambda_i g_{ik}^{(+)}(y_{ik}^*), \tag{2.48}$$

$$k = 1, 2, \ldots, s_i; \quad i = 1, 2, \ldots, m_1,$$

$$\sum_{i=0}^{m_1} \sum_{k=1}^{s_i} \lambda_{ik} a_{ij}^{(k)} = \sum_{j=m_1+1}^{m} \lambda_i a_{ij} \text{ avec } \alpha_j < x_j^* < \beta_j, \tag{2.49}$$

$$\sum_{i=0}^{m_1} \sum_{k=1}^{s_i} \lambda_{ik} a_{ij}^{(k)} \geqslant \sum_{i=m_1+1}^{m} \lambda_i a_{ij} \text{ avec } x_j^* = \alpha_j, \tag{2.50}$$

$$\sum_{i=1}^{m_1} \sum_{k=1}^{s_i} \lambda_{ik} a_{ij}^{(k)} \leqslant \sum_{i=m_1+1}^{m} \lambda_i a_{ij} \text{ avec } x_j^* = \beta_j, \tag{2.51}$$

$$\sum_{i=1}^{m_1} \sum_{k=1}^{s_i} \lambda_i g_{ik}(y_{ik}^*) = 0. \tag{2.52}$$

y_{ik}^* ($k = 1, 2, \ldots, s_i$; $i = 0, 1, \ldots, m_1$) s'expriment ici à l'aide des x_j^* d'après les formules (2.40); $f_k^{(+)}(y_{0k}^*)$, $f_k^{(-)}(y_{0k}^*)$, $g_{ik}^{(+)}(y_{ik}^*)$, $g_{ik}^{(-)}(y_{ik}^*)$ sont les dérivées à droite et à gauche des fonctions f_k et g_{ik} par rapport à y_{0k} et y_{ik} aux points y_{0k}^* et y_{ik}^* respectivement.

La déduction détaillée des conditions (2.47)-(2.52) est laissée au lecteur (cf. exercice 5).

2.5. Des considérations analogues aux précédentes permettent d'énoncer le critère d'optimalité d'un programme pour les autres problèmes particuliers de programmation linéaire par morceaux recensés au paragraphe 1.1.

Ainsi, pour le problème V:

Le programme $X^* = (x_1^*, \ldots, x_n^*)$ du problème V en est une solution si et seulement si il existe un vecteur $\Lambda = (\Lambda_1, \Lambda_2)$ tel qu'il justifie les relations suivantes:

$$f_j^{(-)}(x_j^*) \leqslant (\Lambda_1, A_{\cdot j}) - (\Lambda_2, A_j') \leqslant f_j^{(+)}(x_j^*) \text{ avec } \alpha_j < x_j^* < \beta_j, \tag{2.53}$$

$$f_j^{(+)}(x_j^*) \geqslant (\Lambda_1, A_j) - (\Lambda_2, A_j') \text{ avec } x_j^* = \alpha_j, \tag{2.54}$$

$$f_j^{(-)}(x_j^*) \leqslant (\Lambda_1, A_j) - (\Lambda_2, A_j') \text{ avec } x_j^* = \beta_j, \tag{2.55}$$

$$f_j^{(-)}(y_j^*) \leqslant \lambda_{j-n}^{(2)} \leqslant f_j^{(+)}(y_j^*) \text{ avec } j = n+1, \ldots, n+s. \tag{2.56}$$

Ici, avec $j = n+1, \ldots, n+s$,

$$y_j = L_j(X), \tag{2.57}$$

$$A_j = (a_{1j}, a_{2j}, \ldots, a_{mj})^T;$$

$$A_j' = (a_{n+1,j}, a_{n+2,j}, \ldots, a_{n+s,j})^T, \quad j = 1, 2, \ldots, n,$$

$$\Lambda_1 = (\lambda_1^{(1)}, \ldots, \lambda_m^{(1)}); \quad \Lambda_2 = (\lambda_1^{(2)}, \ldots, \lambda_s^{(2)}).$$

Le critère d'optimalité d'un programme du problème VI s'énonce de la façon suivante.

Pour qu'un programme $X^* = (x_1^*, \ldots, x_n^*)$ du problème VI en soit une solution, il faut et il suffit qu'il existe un vecteur $\Lambda = (\Lambda_1, \Lambda_2)$, avec

$$\Lambda_1 = (\lambda_1^{(1)}, \lambda_2^{(1)}, \ldots, \lambda_m^{(1)}) \geqslant 0, \quad \Lambda_2 = (\lambda_1^{(2)}, \lambda_2^{(2)}, \ldots, \lambda_s^{(2)}),$$

tel qu'il vérifie les relations suivantes:

avec $\alpha_j < x_j^* < \beta_j$,

$$f_j^{(-)}(x_j^*) + (\Lambda_1, g_j^{(-)}(x_j^*)) \leqslant -(\Lambda_2, A_j') \leqslant \leqslant f_j^{(+)}(x_j^*) + (\Lambda_1, g_j^{(+)}(x_j^*)), \tag{2.58}$$

avec $x_j^* = \alpha_j$,

$$f_j^{(+)}(x_j^*) + (\Lambda_1, g_j^{(+)}(x_j^*)) \geqslant -(\Lambda_2, A_j'), \tag{2.59}$$

avec $x_j^* = \beta_j$,

$$f_j^{(-)}(x_j^*) + (\Lambda_1, g_j^{(-)}(x_j^*)) \leqslant -(\Lambda_2, A_j'), \tag{2.60}$$

avec $j = n+1, \ldots, n+s$,

$$f_j^{(-)}(y_j^*) + (\Lambda_1, g_j^{(-)}(y_j^*)) \leqslant \lambda_{j-n}^{(2)} \leqslant \leqslant f_j^{(+)}(y_j^*) + (\Lambda_1, g_j^{(+)}(y_j^*)) \tag{2.61}$$

et

$$\sum_{j=1}^{n} (\Lambda_1, g_j(x_j^*)) + \sum_{j=n+1}^{n+s} (\Lambda_1, g_j(y_j^*)) = 0. \tag{2.62}$$

Ici

$$y_j^* = L_j(X^*), \quad j = n+1, \ldots, n+s,$$

$$A_j' = (a_{n+1,j}, a_{n+2,j}, \ldots, a_{n+s,j})^T, \quad j = 1, 2, \ldots, n,$$

$$g_j(x_j) = \{g_{1j}(x_j), \ldots, g_{mj}(x_j)\}.$$

Les problèmes I à VI sont des formulations particulières du problème VII. Le critère d'optimalité du problème VII peut s'obtenir exactement de la même façon que pour les problèmes V, VI et s'énonce de la façon suivante.

Le programme $X^* = (x_1^*, x_2^*, \ldots, x_n^*)$ du problème VII en est un programme optimal si et seulement si il existe un vecteur $\Lambda = (\Lambda_1, \Lambda_2, \Lambda_3)$, avec

$$\Lambda_1 = (\lambda_1, \ldots, \lambda_{m_1}) \geqslant 0, \quad \Lambda_2 = \lambda_{m_1+1}, \ldots, \lambda_m),$$

$$\Lambda_3 = (\lambda_{m+1}, \ldots, \lambda_{m+s}),$$

tel qu'il vérifie les relations suivantes:

avec $\alpha_j < x_j^* < \beta_j$,

$$f_j^{(-)}(x_j^*) + (\Lambda_1, g_j^{(-)}(x_j^*)) \leqslant (\Lambda_2, A_j) - (\Lambda_3, A_j') \leqslant \leqslant f_j^{(+)}(x_j^*) + (\Lambda_1, g_j^{(+)}(x_j^*)); \tag{2.63}$$

avec $x_j^* = \alpha_j$,

$$f_j^{(+)}(x_j^*) + (\Lambda_1, g_j^{(+)}(x_j^*)) \geqslant (\Lambda_2, A_j) - (\Lambda_3, A_j'); \tag{2.64}$$

avec $x_j^* = \beta_j$,

$$f_j^{(-)}(x_j^*) + (\Lambda_1, g_j^{(-)}(x_j^*)) \leqslant (\Lambda_2, A_j) - (\Lambda_3, A_j'); \tag{2.65}$$

avec $j = n + 1, \ldots, n + s$,

$$f_j^{(-)}(y_j^*) + (\Lambda_1, g_j^{(-)}(y_j^*)) \leqslant \lambda_{j-n}^{(3)} \leqslant$$
$$\leqslant f_j^{(+)}(y_j^*) + (\Lambda_1, g_j^{(+)}(y_j^*)) \tag{2.66}$$

et

$$\sum_{j=1}^{n} (\Lambda_1, g_j(x_j^*)) + \sum_{j=n+1}^{n+s} (\Lambda_1, g_j(y_j^*)) = 0. \tag{2.67}$$

Les notations adoptées ici sont les mêmes que dans le cas précédent. Le lecteur peut voir sans peine que le critère d'optimalité du programme du problème VII permet de déduire aisément tous les critères mentionnés ci-dessus pour les problèmes I à VI (cf. exercice 6).

Les critères d'optimalité des problèmes particuliers de programmation linéaire par morceaux obtenus ici seront employés dans les paragraphes qui suivent à la construction de méthodes et d'algorithmes de calcul du programme optimal.

§ 3. Principes théoriques de la méthode de résolution du problème I

3.1. Illustrons dans le détail, sur l'exemple d'un problème particulier important appelé au § 1 problème I ou problème (B), l'application des critères d'optimalité d'un programme obtenus plus tôt à la construction de méthodes de calcul numérique des problèmes linéaires par morceaux. Les méthodes de résolution d'autres problèmes de programmation linéaire par morceaux se construisent selon un schéma analogue, mais plus complexe (cf. § 7).

Le problème (B) consiste à minimiser la fonction

$$F(X) = \sum_{j=1}^{n} f_j(x_j) \tag{3.1}$$

sous les conditions

$$\sum_{j=1}^{n} A_j x_j = B, \tag{3.2}$$

$$\alpha_j \leqslant x_j \leqslant \beta_j, \quad j = 1, 2, \ldots, n. \tag{3.3}$$

$f_j(x_j)$ sont ici les fonctions linéaires par morceaux convexes vers le bas, associées aux paramètres du problème linéaire équivalent (A) par les relations (1.38).

Posons, de même qu'au § 1,

$$c_{0j} = c_j,\ j = 1,\ 2,\ \ldots,\ n, \tag{3.4}$$

$$c_{ij} = c_j + \sum_{\mu=1}^{i} \sigma_{\mu j}, \quad i = 1, 2, \ldots, l_j; \quad j = 1, 2, \ldots, n. \tag{3.5}$$

Dans ces notations

$$f_j(x_j) = \begin{cases} c_{0j}x_j, & \alpha_j \leqslant x_j \leqslant d_{1j}, \\ c_{1j}(x_j - d_{1j}) + f_j(d_{1j}), & d_{1j} \leqslant x_j \leqslant d_{2j}, \\ \ldots\ldots\ldots\ldots\ldots\ldots & \ldots\ldots \\ c_{l_j j}(x_j - d_{l_j j}) + f_j(d_{l_j j}), & d_{l_j j} \leqslant x_j \leqslant \beta_j. \end{cases} \tag{3.6}$$

D'après le critère d'optimalité du programme d'un problème de type (3.1)-(3.3), obtenu au paragraphe 2.2, le programme $X^* = (x_1^*, \ldots, x_n^*)$ en est une solution si et seulement si il existe un vecteur $\Lambda = (\lambda_1, \ldots, \lambda_m)$ tel qu'il satisfait aux inégalités (2.28)-(2.30). Compte tenu de la définition (3.6) de la fonction $f_j(x_j)$, on peut récrire le critère d'optimalité d'un programme du problème (B) sous la forme suivante.

Pour qu'un programme $X^* = (x_1^*, \ldots, x_n^*)$ du problème (3.1)-(3.3) soit optimal, il faut et il suffit qu'il existe un vecteur $\Lambda = (\lambda_1, \ldots, \lambda_m)$ tel que

$$c_{k-1,\,j} \leqslant \sum_{i=1}^{m} \alpha_{ij}\lambda_i \leqslant c_{kj} \text{ si } x_j^* = d_{kj} \quad (1 \leqslant k \leqslant l_j), \tag{3.7}$$

$$\sum_{i=1}^{m} a_{ij}\lambda_i \leqslant c_{0j} = c_j \text{ si } x_j^* = \alpha_j = d_{0j}, \tag{3.8}$$

$$\sum_{i=1}^{m} a_{ij}\lambda_i \geqslant c_{l_j j} \text{ si } x_j^* = \beta_j = d_{l_j+1,\,j}, \tag{3.9}$$

$$\sum_{i=1}^{m} a_{ij}\lambda_i = c_{kj} \text{ si } d_{kj} < x_j^* < d_{k+1,\,j} \quad (0 \leqslant k \leqslant l_j). \tag{3.10}$$

3.2. De même que dans les problèmes de programmation linéaire, appelons les vecteurs A_j vecteurs des conditions et le vecteur B, vecteur des contraintes. Considérons le rang de la matrice $A = \| a_{ij} \|$ comme égal à m, nombre d'équations du système (3.2).

Les points $\alpha_j = d_{0j}, d_{1j}, \ldots, d_{l_j j}, \beta_j = d_{l_j+1,\,j}$ en lesquels la linéarité de la fonction $f_j(x_j)$ est compromise seront dits *valeurs critiques* de l'argument de cette fonction (valeurs critiques du j-ième argument).

Soit $X = (x_1, x_2, \ldots, x_n)$ un certain programme du problème B. Désignons par $\overline{E}_X$ l'ensemble des indices j tels que x_j ne coïncide

avec aucune de ses valeurs critiques

$$d_{0j},\ d_{1j},\ \ldots,\ d_{l_j+1,\,j}.$$

Nous dirons que le programme $X = (x_1, x_2, \ldots, x_n)$ est *de base* si le système des vecteurs des conditions A_j, avec $j \in \bar{E}_X$, est linéairement indépendant. Le programme de base X est dit *non dégénéré* si l'ensemble $\bar{E}_X$ se compose juste de m éléments.

Nous dirons que le problème du type (B) est *non dégénéré* si chacun de ses programmes de base est non dégénéré. Un système de m vecteurs linéairement indépendants s'appelle *base du programme de base* X s'il contient tous les vecteurs des conditions A_j, avec $j \in \bar{E}_X$. Nous désignerons par E_X l'ensemble des indices des vecteurs des conditions formant une base du programme X. Il est clair qu'un programme non dégénéré n'a qu'une seule base ($E_X = \bar{E}_X$). Les programmes de base dégénérés ($E_X \supset \bar{E}_X$) peuvent posséder plusieurs bases. Nous nous bornerons à l'examen de problèmes non dégénérés et énoncerons dans ce qui suit (§ 4) les considérations qui autorisent l'extension de la méthode aux problèmes dégénérés.

Partons d'un certain programme de base du problème étudié, dont le calcul (s'il faut le faire) peut s'effectuer à l'aide d'une des méthodes de programmation linéaire. La méthode proposée ci-dessous est une généralisation du deuxième algorithme de la méthode primale du simplexe (cf. [87], chapitre 8, § 4).

L'application de cette méthode conduit, comme dans le cas linéaire, à un nombre fini d'itérations de même type. Chaque itération est composée de deux étapes. A la première étape, on teste l'optimalité du programme de base obtenu au terme de l'exécution de l'itération précédente. Si le programme est optimal, la procédure s'arrête. Dans le cas contraire, on effectue la deuxième étape de l'itération, à laquelle l'on construit un nouveau programme de base lié à une plus petite valeur de la fonction $F(X)$ (dans le cas d'une situation non dégénérée), ou bien l'on établit que la fonction $F(X)$ est infinie sur l'ensemble des programmes du problème étudié.

3.3. Voici la description en détail d'une itération distincte de cette méthode. Soit $X = (x_1, x_2, \ldots, x_n)$ un programme de base du problème (B). Pour alléger l'écriture, supposons que sa base se compose de m premiers vecteurs des conditions:

$$A_1,\ A_2,\ \ldots,\ A_m.$$

Etant donné que le problème (B) est supposé non dégénéré, les composantes de base du programme X ne coïncident avec aucune de ses valeurs critiques:

$$d_{k_j j} < x_j < d_{k_j+1,\,j}, \quad 1 \leqslant j \leqslant m. \tag{3.11}$$

En ce qui concerne les composantes secondaires du programme X, chacune d'elles est égale à l'une de ses valeurs critiques:

$$x_j = d_{k_j j}, \quad m+1 \leqslant j \leqslant n. \tag{3.12}$$

Etablissons d'abord si le programme X constitue la solution recherchée du problème. A cet effet essayons de définir d'après le critère d'optimalité, le vecteur $\Lambda = (\lambda_1, \lambda_2, \ldots, \lambda_m)$ associé au programme X par les conditions (3.7)-(3.10). Si un tel vecteur Λ existe, selon la condition (3.10) ses composantes doivent satisfaire au système d'équations

$$\sum_{i=1}^{m} a_{ij}\lambda_i = c_{k_j j}, \quad j = 1, 2, \ldots, m \tag{3.13}$$

Le système (3.13) peut être mis également sous la forme matricielle suivante:

$$\Lambda A_X = C_X, \tag{3.14}$$

où A_X désigne la matrice dont les colonnes sont les vecteurs des conditions qui constituent la base du programme X,

$$C_X = (c_{k_1 1}, c_{k_2 2}, \ldots, c_{k_m m}).$$

Les vecteurs $A_1, A_2, \ldots, A_m$ qui constituent la base du programme X étant linéairement indépendants, le système a une solution unique qui peut s'écrire

$$(\lambda_1, \lambda_2, \ldots, \lambda_m) = \Lambda = C_X A_X^{-1}, \tag{3.15}$$

où A_X^{-1} désigne la matrice inverse de la matrice A_X. D'après le critère d'optimalité, pour que le programme X soit optimal, il faut et il suffit que les composantes ainsi calculées du vecteur Λ vérifient les conditions

$$c_{k_j-1,\,j} \leqslant z_j = \sum_{i=1}^{m} a_{ij}\lambda_i \leqslant c_{k_j j} \text{ si } x_j \neq \alpha_j, \beta_j; \quad j \geqslant m+1,$$

$$c_{l_j j} \leqslant z_j \text{ si } x_j = \beta_j, \quad j \geqslant m+1,$$

$$z_j \leqslant c_{0j} = c_j \text{ si } x_j = \alpha_j, \quad j \geqslant m+1.$$

Si l'on pose par convention $c_{-1j} = -\infty$, $c_{l_j+1,j} = \infty$, tous les trois types de conditions peuvent s'écrire alors sous la forme unique:

$$c_{k_j-1,\,j} \leqslant z_j \leqslant c_{k_j j} \text{ pour } j \geqslant m+1. \tag{3.16}$$

Ainsi, pour tester l'optimalité du programme X, il suffit de calculer les nombres z_j, $j \geqslant m+1$ et d'établir s'ils satisfont au système d'inégalités (3.16). Si toutes les inégalités (3.16) sont vérifiées, alors le programme X est optimal et le problème est résolu.

S'il n'en est pas ainsi, X n'est pas le programme recherché et la procédure doit être poursuivie.

Supposons que nous sommes en présence de la deuxième éventualité, c'est-à-dire qu'au moins une des inégalités (3.16) ne soit pas vérifiée. Cela signifie que parmi les nombres

$$\overline{\Delta}_{k_j j} = c_{k_j j} - z_j, \quad k_j \leqslant l_j, \quad j \geqslant m+1,$$

$$\overline{\overline{\Delta}}_{k_j j} = z_j - c_{k_j - 1, j}, \quad k_j \geqslant 1, \quad j \geqslant m+1,$$

il en est qui sont négatifs.

Remarquons qu'en raison de la définition des nombres c_{ij},

$$\overline{\overline{\Delta}}_{ij} = -\overline{\Delta}_{ij} + \sigma_{ij} \quad \text{avec} \quad 1 \leqslant i \leqslant l_j. \tag{3.17}$$

Choisissons un indice $j = r$ pour lequel est enfreinte l'une des inégalités de la j-ième ligne du système (3.16). Deux cas sont possibles, qui s'excluent mutuellement :

1°. $\overline{\Delta}_{k_r r} < 0$ (c'est le deuxième membre de la r-ième inégalité qui est compromis).

2°. $\overline{\overline{\Delta}}_{k_r r} < 0$ (c'est le premier membre de la r-ième inégalité qui est compromis).

Considérons séparément chacun de ces cas.

1°. $\Delta = \overline{\Delta}_{k_r r} = c_{k_r r} - z_r < 0$.

Décomposons le vecteur A_r par rapport à la base du programme X :

$$A_r = \sum_{i=1}^{m} x_{ir} A_i. \tag{3.18}$$

Notre tâche consiste à construire un nouveau programme de base du problème B, associé à une plus petite valeur de la fonction $F(X)$. Introduisons dans la discussion le vecteur $X(\theta) = (x_1(\theta), \ldots, x_n(\theta))$ dont les composantes se calculent d'après la formule

$$x_j(\theta) = \begin{cases} x_j - \theta x_{jr}, & j = 1, 2, \ldots, m, \\ x_r + \theta, & j = r, \\ x_j, & j \neq r, \quad j \geqslant m+1. \end{cases} \tag{3.19}$$

Notons que pour une valeur quelconque de θ,

$$\sum_{j=1}^{n} A_j x_j(\theta) = \sum_{j=1}^{n} A_j x_j + \theta \left[A_r - \sum_{i=1}^{m} x_{ir} A_i\right].$$

On en tire, compte tenu de l'égalité (3.18),

$$\sum_{j=1}^{n} A_j x_j(\theta) = \sum_{j=1}^{n} A_j x_j = B,$$

c'est-à-dire le vecteur $X(\theta)$ satisfait aux contraintes (3.2).

Si $\theta > 0$ est suffisamment petit, les composantes du vecteur $X(\theta)$ vérifient également les conditions (3.3) du problème considéré. En effet, avec $1 \leqslant j \leqslant m$, ceci résulte des inégalités (3.11). Comme c'est le cas 1° qui a lieu, $x_r < \beta_r$, d'où $\alpha_r \leqslant x_r + \theta \leqslant \beta_r$ pour θ suffisamment petit.

Les autres composantes du vecteur $X(\theta)$ coïncident avec les composantes respectives du programme X, qui justifient les contraintes (3.3).

Ainsi, avec des θ positifs suffisamment petits, le vecteur $X(\theta)$ est un programme du problème (B). La borne supérieure θ' de ces valeurs de θ se calcule sans peine.

Désignons par E'_X (E''_X) l'ensemble des indices $j \in E_X$ tels que

$$x_{jr} > 0 \quad (x_{jr} < 0).$$

θ' est évidemment le plus petit des trois nombres:

$$\min_{j \in E'_X} \frac{x_j - \alpha_j}{x_{jr}}, \quad \min_{j \in E''_X} \frac{x_j - \beta_j}{x_{jr}}, \quad \beta_r - x_r. \tag{3.20}$$

Observons maintenant comment change la valeur de la fonction à minimiser $F(X)$ à la suite du passage à un nouveau programme $X(\theta)$. A cet effet réalisons la construction auxiliaire suivante. Lorsque θ varie de 0 à θ', les composantes du programme $X(\theta)$ de numéros $1, 2, \ldots, m, r$ peuvent passer par leurs valeurs critiques (différentes, certes, de α_j et β_j). Ecrivons toutes celles des valeurs de θ $(0 \leqslant \theta \leqslant \theta')$ qui peuvent donner lieu à cette situation. La composante $x_j(\theta)$, $1 \leqslant j \leqslant m$, prend la valeur $d_{\gamma j}$ avec

$$\theta = \frac{x_j - d_{\gamma j}}{x_{jr}}. \tag{3.21}$$

Si l'on pose $x_{rr} = -1$, la relation (3.21) reste vraie aussi pour $j = r$. On en déduit que les valeurs de θ qui nous intéressent peuvent s'écrire

$$\theta_{\gamma j} = \frac{x_j - d_{\gamma j}}{x_{jr}} \quad \text{pour } j = 1, 2, \ldots, m, r$$

et pour des $\gamma \neq 0, l_j + 1$, tels que $0 \leqslant \theta_{\gamma j} \leqslant \theta'$.

On voit donc que $j \in E'_X$ tient compte des valeurs critiques $d_{\gamma j}$ situées à gauche de x_j et $j \in E''_X$ et $j = r$, des valeurs critiques à droite de x_j. On ne prend ici en considération que celles de ces valeurs qui peuvent être obtenues avec $\theta \leqslant \theta'$ et qui diffèrent de α_j et β_j. Ecrivons les valeurs trouvées dans l'ordre croissant:

$$\theta_{\gamma_1 j_1} \leqslant \theta_{\gamma_2 j_2} \leqslant \ldots \leqslant \theta_{\gamma_s j_s}. \tag{3.22}$$

Constatons que lorsque le problème (B) est non dégénéré, tous les membres de la suite (3.22) sont différents et positifs. (Autrement,

avec un certain θ, l'ensemble $\bar{E}_{X(\theta)}$, pour le programme $X(\theta)$, se composerait d'un nombre d'éléments inférieur à m). Supposons qu'au début θ est inclus dans le segment $[0, \theta_{\gamma_1 j_1}]$. Sous cette condition pour θ, les arguments de la fonction $F(X)$ varient dans les limites

$$d_{k_j j} \leqslant x_j(\theta) \leqslant d_{k_j+1, j}.$$

Compte tenu que d'après la condition la fonction $f_j(x_j)$ est linéaire sur le segment $[d_{k_j j}, d_{k_j+1, j}]$ et que son coefficient angulaire est $c_{k_j j}$, il vient

$$F(X(\theta)) - F(X) = \theta\left[c_{k_r r} - \sum_{j=1}^{m} c_{k_j j} x_{jr}\right]. \tag{3.23}$$

On sait (cf., par exemple, [87], chapitre 7, paragraphe 1.2) que $\sum_{j=1}^{m} c_{k_j j} x_{jr} = z_r$. C'est pourquoi

$$c_{k_r r} - \sum_{j=1}^{m} c_{k_j j} x_{jr} = c_{k_r r} - z_r = \bar{\Delta}_{k_r r} = \Delta.$$

Donc, avec $0 \leqslant \theta \leqslant \theta_{\gamma_1 j_1}$,

$$F(X(\theta)) - F(X) = \Delta\theta < 0,$$

c'est-à-dire le passage au programme $X(\theta)$ fait décroître la fonction $F(X)$ à un rythme d'autant plus rapide que la quantité θ est plus grande.

θ allant en augmentant, l'argument d'une des fonctions f_j passe par sa valeur critique, et l'égalité (3.23) est compromise.

Supposons que $\theta_{\gamma_1 j_1} \leqslant \theta \leqslant \theta_{\gamma_2 j_2}$. θ variant dans l'intervalle considéré, les arguments des fonctions f_j se situent dans les intervalles de linéarité de ces fonctions. De plus, les coefficients angulaires de toutes les fonctions, sauf f_{j_1}, restent les mêmes. Comme nous sommes passés dans l'intervalle voisin de linéarité de la fonction f_{j_1}, son coefficient angulaire devient

$$c_{k_{j_1}+1} \text{ si } x_{j_1 r} < 0 \quad (j_1 \in E_X^{\prime\prime}),$$

$$c_{k_{j_1}-1} \text{ si } x_{j_1 r} > 0 \quad (j_1 \in E_X^{\prime}).$$

Dans le premier cas, $\gamma_1 = k_j + 1$; dans le deuxième, $\gamma_1 = k_j$.

En utilisant les raisonnements qui précèdent, on obtient que pour $\theta_{\gamma_1 j_1} \leqslant \theta \leqslant \theta_{\gamma_2 j_2}$,

$$F(X(\theta)) - F(X) = [F(X(\theta_{\gamma_1 j_1})) - F(X)] + [F(X(\theta)) - F(X(\theta_{\gamma_1 j_1}))] =$$
$$= \Delta\theta_{\gamma_1 j_1} + (\Delta + \sigma_{\gamma_1 j_1} | x_{j_1 r} |)(\theta - \theta_{\gamma_1 j_1}) = \Delta\theta + \sigma_{\gamma_1 j_1} | x_{j_1 r} | (\theta - \theta_{\gamma_1 j_1}).$$

Supposons enfin que θ se situe entre deux termes voisins quelconques de la suite (3.22) :

$$\theta_{\gamma_t j_t} \leqslant \theta \leqslant \theta_{\gamma_{t+1} j_{t+1}} \quad (t = 1, 2, \ldots, s-1).$$

Des raisonnements parfaitement analogues à ceux exposés plus haut conduisent à l'égalité

$$F(X(\theta)) - F(X) = \Delta \cdot \theta + \sum_{i=1}^{t} \sigma_{\gamma_i j_i} |x_{j_i r}| (\theta - \theta_{\gamma_i j_i}). \qquad (3.24)$$

Posons

$$\Delta_t = \Delta + \sum_{i=1}^{t} \sigma_{\gamma_i j_i} |x_{j_i r}| \quad (t = 1, 2, \ldots, s).$$

De la relation (3.24) il suit qu'avec $\theta_{\gamma_t j_t} \leqslant \theta \leqslant \theta_{\gamma_{t+1} j_{t+1}}$ la décroissance de $(F(X(\theta))$, lorsque θ croît, n'a lieu que si $\Delta_t < 0$. Naturellement, augmenter θ n'a de sens que tant que cela conduit à une décroissance de $F(X(\theta))$. Cette condition définit pour θ la deuxième borne supérieure que nous désignerons par θ''. Il est clair que θ'' peut être calculé en se guidant sur la règle suivante.

Si $\Delta_s < 0$, $\theta'' = \infty$.

Si $\Delta_s \geqslant 0$, après avoir trouvé l'entier naturel ν à partir de la condition

$$\Delta_{\nu-1} < 0, \quad \Delta_\nu \geqslant 0, \quad (\Delta_0 = \Delta),$$

posons

$$\theta'' = \theta_{\gamma_\nu j_\nu}.$$

La borne définitive de variation de θ, désignée par θ_0, est égale au plus petit des nombres θ', θ'' :

$$\theta_0 = \min \{\theta', \theta''\}.$$

Notons qu'avec $\Delta_s < 0$ $(\theta'' = \infty)$,

$$\theta_0 = \theta'.$$

Nous dirons par la suite que θ_0 est réalisé sur le vecteur A_j, $j = 1, 2, \ldots, m, r$, si $\theta_0 = -\dfrac{d_{\gamma j} - x_j}{x_{jr}}$ avec un certain γ.

Deux cas peuvent se présenter :

a) $\theta_0 = \infty$;

b) $\theta_0 < \infty$.

Dans le premier cas, $X(\theta)$ est un programme du problème B quel que soit $\theta > 0$, et de plus,

$$F(X(\theta)) - F(X) \leqslant \Delta_s \theta, \quad \Delta_s < 0.$$

Par conséquent, dans le cas considéré, le problème n'a pas de solution. (La fonction $F(X)$ n'est pas bornée inférieurement sur l'ensemble des programmes du problème.)

Constatons que le cas a) est impossible si tous les nombres α_j, β_j sont finis.

Considérons le cas b). D'après le choix du nombre θ_0, le programme $X(\theta_0) = X'$ correspond à la plus petite valeur de la fonction F par rapport à tous les programmes du type (3.19). Ce programme est adopté comme approximation consécutive pour la résolution du problème. On vérifie aisément qu'il est également de base. La base du programme X' se forme à partir de la base du programme X précédent en remplaçant par le vecteur A_r le vecteur A_k sur lequel se réalise la valeur θ_0. Il se peut que $k = r$. Dans ce cas, les programmes X et X' ont la même base. Notons que dans le cas non dégénéré la quantité θ_0 est toujours réalisée sur un des vecteurs des conditions seulement. En effet, s'il y en avait plusieurs, alors, les composantes du programme X' qui leur sont associées étant égales à certaines de leurs propres valeurs critiques, l'ensemble $\bar{E}_{X'}$ contiendrait moins de m éléments, ce qui témoignerait de la dégénérescence du programme X'. La formule (3.24) permet de déduire sans peine l'expression de la variation de la fonction $F(X)$ lors du passage à un nouveau programme X':

$$F(X') - F(X) = \Delta \cdot \theta_0 + \sum_{i=1}^{t} \sigma_{\gamma_i j_i} |x_{j_i r}| (\theta_0 - \theta_{\gamma_i j_i}), \qquad (3.25)$$

où t est défini comme le plus grand indice i tel que

$$\theta_{\gamma_i j_i} \leqslant \theta_0.$$

Puisque $\theta_{\gamma_1 j_1} > 0$ (le programme X est non dégénéré), $\theta_0 \geqslant \theta_{\gamma_1 j_1} > 0$. Par suite, $F(X') - F(X) < 0$.

2°. Considérons maintenant l'autre éventualité, qui consiste en ce que

$$\Delta = \bar{\bar{\Delta}}_{kr r} = z_r - c_{k_r - 1, r} < 0.$$

Dans la mesure où toutes les justifications se rapportant au passage à un nouveau programme de base X' coïncident totalement avec celles qui ont été données déjà lors de la description du cas 1°, nous nous bornerons au seul exposé de l'ordre des opérations.

Introduisons le vecteur $X(\theta)$ dont les composantes se calculent d'après la formule

$$x_j(\theta) = \begin{cases} x_j + \theta x_{jr}, & j = 1, 2, \ldots, m, \\ x_r - \theta, & j = r, \\ x_j, & j \geqslant m + 1, \quad j \neq r. \end{cases} \qquad (3.26)$$

Le nouveau programme est

$$X' = X(\theta_0) \quad \text{si} \quad \theta_0 < \infty.$$

La procédure de calcul de la grandeur θ_0 consiste à définir d'abord θ' comme le minimum de trois nombres :

$$\min_{j \in E'_X} \left(\frac{\beta_j - x_j}{x_{jr}} \right), \quad \min_{j \in E''_X} \left(\frac{\alpha_j - x_j}{x_{jr}} \right), \quad x_r - \alpha_r. \tag{3.27}$$

On calcule ensuite les nombres

$$\theta_{\gamma j} = \frac{d_{\gamma j} - x_j}{x_{jr}}$$

pour $j = 1, 2, \ldots, m, r$ et pour γ tels que

$$0 \leqslant \theta_{\gamma j} \leqslant \theta', \quad \gamma \neq 0, \quad l_j + 1,$$

et on les dispose par ordre de croissance.

Supposons que l'on forme de ce fait la suite

$$\theta_{\gamma_1 j_1} \leqslant \theta_{\gamma_2 j_2} \leqslant \ldots \leqslant \theta_{\gamma_s j_s}. \tag{3.28}$$

Soit

$$\Delta_t = \Delta + \sum_{i=1}^{t} \sigma_{\gamma_i j_i} |x_{j_i r}| \quad (t = 1, 2, \ldots, s).$$

Il est clair que $\Delta = \Delta_0 < \Delta_1 < \ldots < \Delta_s$. Cherchons le nombre θ'' en nous guidant sur la règle suivante :

$$\theta'' = \infty \quad \text{si} \quad \Delta_s < 0.$$

Si $\Delta_s \geqslant 0$, alors $\theta'' = \theta_{\gamma_\nu j_\nu}$, où l'entier naturel ν est défini par la condition

$$\Delta_{\nu-1} < 0, \quad \Delta_\nu \geqslant 0.$$

La valeur recherchée est

$$\theta_0 = \min (\theta', \theta'').$$

De même que dans le premier cas, θ_0 peut être soit fini, soit infini.

Si $\theta_0 = \infty$, le problème B est irréalisable.

Mais si $\theta_0 < \infty$, $X' = X(\theta_0)$ est alors un programme de base du problème, et dans ce cas là aussi on vérifie la formule (3.25) :

$$F(X') - F(X) = \Delta \cdot \theta_0 + \sum_{i=1}^{t} \sigma_{\gamma_i j_i} |x_{j_i r}| (\theta_0 - \theta_{\gamma_i j_i}),$$

où t est le plus grand des indices i justifiant la condition

$$\theta_{\gamma_i j_i} \leqslant \theta_0.$$

Dans le cas non dégénéré considéré $\theta_0 > 0$, et, par conséquent,

$$F(X') - F(X) < 0.$$

La construction du programme X' achève la deuxième étape de l'itération et l'itération elle-même.

X' est le programme initial pour l'exécution de l'itération suivante. Les itérations successives se poursuivent tant qu'on établisse que la fonction à minimiser (3.1) est non bornée inférieurement, ou bien tant qu'on obtienne la solution du problème. Dans le cas non dégénéré, chaque itération ainsi qu'on l'a déjà noté, conduit à une diminution de la fonction $F(X)$. Il en résulte que l'exploration du problème (B) s'effectue en un nombre fini d'itérations. En effet, chaque itération (sauf la dernière) s'achève par la construction d'un programme de base. La décroissance monotone de la fonction $F(X)$ fait que tous ces programmes sont différents. Comme, d'autre part, le nombre de bases du problème ne dépasse pas C_n^m, et comme le nombre de programmes de base $(l = \max\limits_{1 \leq j \leq n} l_j)$ associés à chaque base n'est pas supérieur à $(l+2)^{n-m}$, le nombre total de programmes de base du problème (B) est borné par le nombre $(l+2)^{n-m} \cdot C_n^m$.

Ainsi, la procédure de résolution d'un problème non dégénéré du type (B) ne peut pas compter plus de $(l+2)^{n-m} \cdot C_n^m$ itérations. Constatons que, pratiquement, ce nombre est nettement inférieur au critère grossier donné ici.

§ 4. Dégénérescence

Dans le paragraphe précédent, nous nous sommes bornés à examiner les problèmes non dégénérés du type (B). Nous allons maintenant étendre la méthode décrite auparavant au cas général.

Si tous les programmes du problème (B) sont non dégénérés, chaque itération conduit à une décroissance de la fonction $F(X)$, puisque dans ce cas $\theta_0 > 0$. Dans le cas dégénéré, la quantité θ_0 peut être égale à zéro, et l'itération respective ne réduit pas la valeur de la fonction à minimiser. Rappelons que la démonstration de la finitude de la méthode était étayée essentiellement par le fait que le processus était monotone, puisque c'est précisément cette condition qui garantit que les programmes de base construits au cours de la résolution du problème sont différents. La dégénérescence du problème se manifeste généralement par le fait que la quantité θ_0 s'obtient simultanément sur plusieurs vecteurs des conditions. On pourrait, certes, éliminer de la base l'un quelconque de ces vecteurs. Alors, comme dans le cas non dégénéré, nous nous déplacerions suivant les programmes de base du problème.

Toutefois, la fonction $F(X)$ pouvant dans ces conditions garder une valeur constante, il se peut que deux programmes séparés par plusieurs itérations coïncident. Ceci peut conduire à son tour à ce que l'on appelle un cyclage qui consiste en un retour périodique au même programme de base après quelques itérations. Des exemples montrent que l'apparition d'un cyclage est possible. Pour parer à cette éventualité, il faut compléter la méthode par une règle

d'élimination de la base des vecteurs des conditions qui rendrait impossible la formation, au cours de la procédure de résolution, de deux programmes de base identiques.

Introduisons dans la discussion, au lieu du problème (B), le ε-problème suivant.

Minimiser la fonction

$$F(X)=\sum_{j=1}^{n} f_j(x_j) \tag{4.1}$$

sous les conditions

$$\sum_{j=1}^{n} A_j x_j = B + \sum_{k=1}^{m} \varepsilon^k R_k, \tag{4.2}$$

$$\alpha_j \leqslant x_j \leqslant \beta_j, \quad j = 1, 2, \ldots, n, \tag{4.3}$$

où R_k $(k = 1, 2, \ldots, m)$ est un système de vecteurs de dimension m;

ε, un nombre arbitraire non négatif.

Appelons (4.1)-(4.3) problème

$$B\ (\varepsilon,\ R_1,\ R_2,\ \ldots,\ R_m).$$

Le nouveau problème ne diffère de (B) que par les contraintes (4.2) qui remplacent dans la formulation les conditions (3.2). La justification donnée plus bas de la règle supplémentaire de choix du vecteur des conditions à retirer de la base s'appuie sur deux propositions, dont la démonstration est laissée au lecteur (cf. exercice 7).

1. *Si le système de vecteurs $R_1, R_2, \ldots, R_m$ est linéairement indépendant, il existe un $\varepsilon' > 0$ tel qu'avec*

$$0 < \varepsilon \leqslant \varepsilon' \tag{4.4}$$

le problème $B\ (\varepsilon,\ R_1,\ R_2,\ \ldots,\ R_m)$ soit non dégénéré.

2. *Il existe un $\varepsilon'' > 0$ tel que si*

$$0 < \varepsilon \leqslant \varepsilon'' \tag{4.5}$$

et si $X\ (\varepsilon) = (x_1\ (\varepsilon), \ldots, x_n\ (\varepsilon))$ est un programme de base quelconque du problème $B\ (\varepsilon,\ R_1,\ R_2,\ \ldots,\ R_m)$, alors $X\ (0) = X = (x_1,\ x_2,\ \ldots,\ x_n)$ est un programme de base du problème (B) avec la même base, les composantes de base $x_j\ (\varepsilon)$ et $x_j\ (0)$ se situant de plus entre les mêmes valeurs critiques. Si $X\ (\varepsilon)$ est le programme optimal du problème $B\ (\varepsilon,\ R_1,\ R_2,\ \ldots,\ R_m)$, X est alors la solution du problème (B).

Supposons que la procédure de résolution du problème (B) débute par un certain programme de base $X = (x_1,\ x_2,\ \ldots,\ x_n)$ ayant pour base le système de vecteurs $A_1,\ A_2,\ \ldots,\ A_m$.

Considérons le problème (B) simultanément avec le problème $B\ (\varepsilon,\ R_1,\ R_2,\ \ldots,\ R_m)$, où ε vérifie les inégalités (4.4) et (4.5)

et où $R_1, R_2, \ldots, R_m$ est un système de vecteurs linéairement indépendant.

Supposons que le vecteur $X(\varepsilon)$, où $x_j(\varepsilon) = x_j$ avec $j \geqslant m+1$, soit un programme de base du problème $B(\varepsilon, R_1, R_2, \ldots, R_m)$ et ait par conséquent pour base le système $A_1, A_2, \ldots, A_m$.

Cherchons à résoudre le problème $B(\varepsilon, R_1, R_2, \ldots, R_m)$ par la méthode décrite au paragraphe précédent, en partant du programme $X(\varepsilon)$. Le problème $B(\varepsilon, R_1, R_2, \ldots, R_m)$ étant non dégénéré en vertu de la proposition 1, nous construisons sa solution $\widetilde{X}(\varepsilon)$, après avoir passé par les programmes $X^{(1)}(\varepsilon), \ldots, X^{(s)}(\varepsilon)$, en un nombre fini d'itérations.

D'après la proposition 2, les vecteurs $X^{(i)}(0) = X^{(i)}$, $\widetilde{X}(0) = \widetilde{X}$ sont des programmes de base du problème (B), et de plus, $\widetilde{X}$ est la solution de ce problème. La procédure de résolution du problème $B(\varepsilon, R_1, R_2, \ldots, R_m)$ se réduit ainsi à se déplacer suivant les programmes de base correspondants du problème (B), pour aboutir, après un nombre fini d'itérations, à la solution du problème (B).

Suivons ce cheminement sur l'exemple de passage du programme $X = X(0)$ au programme $X^{(1)} = X^{(1)}(0)$. En vertu du choix de ε (proposition 2), les composantes correspondantes $X(\varepsilon)$ et X sont comprises entre les mêmes valeurs critiques. Aussi peut-on effectuer la première étape de l'itération en partant du programme X. Supposons que le programme X ne satisfasse pas au critère d'optimalité et que le vecteur des conditions à introduire dans la nouvelle base soit A_r.

En analysant la deuxième étape, nous distinguerons deux cas, de même que dans la description de la méthode.

1°. Dans le premier cas, le passage au nouveau programme se fait d'après les formules (3.19). Pour calculer $\theta_0 = \min(\theta', \theta'')$, il faut comparer entre elles les quantités $\theta_{\gamma j}(\varepsilon)$, $0 \leqslant \gamma \leqslant l_j + 1$, qui dans le cas non dégénéré sont toujours différentes. Ceci étant, on élimine de la base le vecteur A_k sur lequel s'obtient la quantité θ_0. L'indice k est ainsi défini par la condition : avec un certain γ,

$$\theta_{\gamma k}(\varepsilon) = \theta_0.$$

Pour n'importe quels γ et $j \in E = (1, 2, \ldots, m)$, $x_{jr} \neq 0$,

$$\theta_{\gamma j}(\varepsilon) = \frac{x_j(\varepsilon) - d_{\gamma j}}{x_{jr}} = \frac{x_j + \sum_{s=1}^{m} \varepsilon^s \bar{r}_{js} - d_{\gamma j}}{x_{jr}} = \theta_{\gamma j} + \sum_{s=1}^{m} \varepsilon^s \frac{\bar{r}_{js}}{x_{jr}}. \quad (4.6)$$

où $\bar{r}_{1s}, \bar{r}_{2s}, \ldots, \bar{r}_{ms}$ désignent les coefficients de décomposition du vecteur R_s par rapport au système qui forme la base du pro-

gramme $X(\varepsilon)$. Si r est le numéro du vecteur introduit,

$$\theta_{\gamma r}(\varepsilon) = \theta_{\gamma r}. \tag{4.7}$$

Etant donné que ε peut être considéré comme un nombre aussi petit que l'on veut, la formule (4.6) et l'égalité (4.7) conduisent à la règle suivante de comparaison des quantités $\theta_{\gamma j}$:

$$\theta_{\gamma_1 j_1} \text{ est « inférieur » à } \theta_{\gamma_2 j_2} \text{ si } \theta_{\gamma_1 j_1} < \theta_{\gamma_2 j_2}$$

ou

$$\theta_{\gamma_1 j_1} = \theta_{\gamma_2 j_2}, \quad \frac{\bar{r}_{j_1 s}}{x_{j_1 r}} = \frac{\bar{r}_{j_2 s}}{x_{j_2 r}} \text{ pour } s = 1, 2, \ldots, t,$$

mais

$$\frac{\bar{r}_{j_1, t+1}}{x_{j_1 r}} < \frac{\bar{r}_{j_2, t+1}}{x_{j_2 r}}.$$

Ici, par définition, $\bar{r}_{rs} = 0$ pour $s = 1, 2, \ldots, m$.

Ainsi, la deuxième étape de l'itération peut être réalisée également en partant du programme X, à condition d'appliquer pour comparer les quantités $\theta_{\gamma j}$ identiques la règle qui vient d'être énoncée.

Supposons que θ_0 s'obtient sur plusieurs vecteurs des conditions: $\theta_0 = \theta_{\gamma_\nu j_\nu} = \theta_{\gamma_{\nu+1} j_{\nu+1}} = \ldots = \theta_{\gamma_{\nu+d} j_{\nu+d}}$.

En nous servant de la règle indiquée ci-dessus, nous rangerons les quantités $\theta_{\gamma_{\nu+i} j_{\nu+i}}$ par l'ordre de « croissance ». Supposons que la « plus petite » soit $\theta_{\gamma_{\nu+\delta} j_{\nu+\delta}}$. Alors, l'indice k du vecteur A_k à éliminer de la base est défini par l'égalité

$$k = j_{\nu+\delta}.$$

Constatons que les composantes de base du programme $X^{(1)}$, égales aux valeurs critiques des arguments correspondants, sont considérées de la même manière que si elles n'avaient pas atteint ces valeurs critiques. De cette façon, si le vecteur A_j entre dans les bases des programmes X et $X^{(1)}$ et si, de plus, $x_j^{(1)} = d_{\gamma j}$ avec un certain γ, on considère alors $x_j^{(1)}$ comme situé soit sur $[d_{\gamma-1, j}, d_{\gamma j}]$, soit sur $[d_{\gamma j}, d_{\gamma+1, j}]$, suivant que le segment de linéarité de la fonction $f_j(x_j)$ contenant x_j se trouve à gauche ou à droite de $d_{\gamma j}$. Si le vecteur A_r ajouté reste dans la base du programme $X^{(1)}$ et si $x_r^{(1)} = d_{\gamma r}$, on considère que $x_r^{(1)}$ repose sur $[d_{\gamma-1, r}, d_{\gamma r}]$, puisque dans le cas considéré $x_r^{(1)} = x_r + \theta_0$.

2°. Dans le deuxième cas, pour passer au nouveau programme $X^{(1)}$ on fait appel aux formules (3.26). Etant donné maintenant que

$$\theta_{\gamma j}(\varepsilon) = \frac{d_{\gamma j} - x_j(\varepsilon)}{x_{jr}} = \theta_{\gamma j} - \sum_{s=1}^{m} \varepsilon^{(s)} \frac{\bar{r}_{js}}{x_{jr}}, \tag{4.8}$$

$$j = 1, 2, \ldots, m; \quad \theta_{\gamma r}(\varepsilon) = \theta_{\gamma r},$$

la règle nécessaire pour comparer les valeurs de $\theta_{\gamma j}$ identiques a la forme suivante.

Soit $\theta_{\gamma_1 j_1} = \theta_{\gamma_2 j_2}$. La grandeur $\theta_{\gamma_1 j_1}$ est considérée comme « inférieure » à $\theta_{\gamma_2 j_2}$ si

$$\frac{\bar{r}_{j_1 s}}{x_{j_1 r}} = \frac{\bar{r}_{j_2 s}}{x_{j_2 r}} \text{ pour } s = 1, 2, \ldots, t, \text{ mais } \frac{\bar{r}_{j_1, t+1}}{x_{j_1 r}} > \frac{\bar{r}_{j_2, t+1}}{x_{j_2 r}},$$

où, de même que dans le premier cas, $\bar{r}_{rs} = 0$ pour $s = 1, 2, \ldots, m$. Le vecteur à retrancher de la base est défini de la même façon que dans le premier cas, à l'aide de la règle établie ci-dessus.

Comme dans le premier cas, les composantes de base du programme $X^{(1)}$ égales aux valeurs critiques sont considérées comme n'ayant pas atteint ces valeurs. C'est pourquoi, si A_j entre dans les bases des programmes X et $X^{(1)}$, la règle de détermination du segment de linéarité de la fonction $f_j(x_j)$ contenant $x_j^{(1)}$ reste inchangée. Si le vecteur A_r à introduire reste dans la nouvelle base et si, de plus, $x_r^{(1)} = d_{\gamma r}$, $x_r^{(1)}$ est alors considéré comme situé sur le segment $[d_{\gamma r}, d_{\gamma+1, r}]$.

Ainsi, pour appliquer la méthode examinée rendant impossible le cyclage, il faut disposer de la décomposition des vecteurs d'un certain système linéairement indépendant par rapport aux vecteurs de la base courante. La forme concrète des vecteurs $R_1, R_2, \ldots, R_m$ de ce système est indiquée dans le paragraphe suivant.

§ 5. Algorithme de programmation linéaire par morceaux

5.1. Nous avons déjà dit que la résolution du problème (B) débute par la détermination de l'un de ses programmes de base. Ce dernier peut s'obtenir en recourant à l'une quelconque des méthodes de programmation linéaire. Il arrive souvent que la recherche du programme initial n'impose pas l'utilisation de la programmation linéaire, du fait que la construction d'un tel programme est parfaitement évidente. Supposons que nous connaissons un certain programme de base

$$X = (x_1, x_2, \ldots, x_n),$$

à base $A_{j_1}, A_{j_2}, \ldots, A_{j_m}$, et la matrice A_X^{-1} inverse de la matrice $A_X = (A_{j_1}, A_{j_2}, \ldots, A_{j_m})$. Transformons les contraintes du problème

$$\sum_{j=1}^{n} A_j x_j = B$$

en multipliant les deux membres par A_X^{-1}.

Si l'on pose $A_X^{-1}A_j' = A_j'$, $A_X^{-1}B = B'$, ces contraintes se mettent sous la forme

$$\sum_{j=1}^{n} A_j' x_j = B',$$

et, de plus, pour $s = 1, 2, \ldots, m$,

$$A_{j_s}' = e_s,$$

où $e_s = (\underbrace{0, 0, \ldots, 0, 1}_{s}, 0, \ldots, 0)$ est le s-ième vecteur unité.

Pour simplifier l'écriture, considérons que ces transformations sont déjà accomplies. De la sorte, la base du programme initial X se compose de vecteurs unités. Au cours de la résolution du problème, nous effectuerons la décomposition des vecteurs $A_{j_s}' = e_s$, $s = 1, 2, \ldots, m$, par rapport aux vecteurs d'une base courante quelconque. On peut profiter de cette circonstance pour obtenir une réalisation économique de la règle exposée au paragraphe précédent et rendant impossible le cyclage.

Posons pour $k = 1, 2, \ldots, m$

$$R_k = \begin{cases} e_k & \text{si } x_{j_k} < \beta_{j_k}, \\ -e_k & \text{si } x_{j_k} = \beta_{j_k}. \end{cases} \tag{5.1}$$

Il est clair que le système $R_1, R_2, \ldots, R_m$ est linéairement indépendant et que la base du programme X est également la base du programme $X(\varepsilon)$ du problème $B(\varepsilon, R_1, \ldots, R_m)$ avec $\varepsilon > 0$ suffisamment petit. Cette dernière proposition est justifiée par le type des vecteurs $R_1, R_2, \ldots, R_m$. Nous utiliserons le système $R_1, R_2, \ldots, R_m$ ainsi choisi pour former la règle d'élimination des vecteurs de la base courante.

5.2. Voici maintenant la description du schéma de calcul de la méthode.

Il est commode de ranger les conditions du problème en deux tableaux. Dans le tableau I (tableau 7.1) on inscrit les composantes des vecteurs des conditions A_j $(j = 1, 2, \ldots, n)$ et le vecteur des contraintes B.

Dans le tableau II (tableau 7.2) on porte les valeurs critiques des arguments de la fonction $F(X)$, les grandeurs σ_{ij} et c_{ij}.

Les colonnes du tableau II sont évidemment de longueur différente. Celles qui correspondent à la variable x_j (groupe de colonnes j) contiennent chacune $l_j + 2$ éléments. C'est pourquoi nous n'avons tracé que deux lignes du tableau.

Décrivons une itération élémentaire (la l-ième) de l'algorithme. Les données de départ de cette itération sont le programme de base

$$X = (x_1, x_2, \ldots, x_n),$$

Tableau 7.1

I

$i \backslash A_j$	A_1	A_2	.....	A_n	B
1	a_{11}	a_{12}	.....	a_{1n}	b_1
2	a_{21}	a_{22}	.....	a_{2n}	b_2
....	... ...	... ...	....	... ...	... ...
$m-1$	$a_{m-1,1}$	$a_{m-1,2}$	.....	$a_{m-1,n}$	b_{m-1}
m	a_{m1}	a_{m2}	.....	a_{mn}	b_m

dont la base se compose des vecteurs $A_1, A_2, \ldots, A_m$ *) et la matrice $A_X^{-1} = (A_1, A_2, \ldots, A_m)^{-1}$. Ces données sont portées sur le tableau 7.3 que nous appellerons par la suite tableau principal de l'itération.

Tableau 7.2

II

$\gamma \backslash j$	1			2			...	n		
—	d_{k1}	σ_{k1}	c_{k1}	d_{k2}	σ_{k2}	c_{k2}	...	d_{kn}	σ_{kn}	c_{kn}
0	α_1	—	c_{01}	α_2	—	c_{02}	...	α_n	—	c_{0n}
1	d_{11}	σ_{11}	c_{11}	d_{12}	σ_{12}	c_{12}	...	d_{1n}	σ_{1n}	c_{1n}
. . . . .	. . . . .	. . . . .	. . . . .	. . . . .	. . . . .	. . . . .	... ...	. . . . .	. . . . .	. . . . .

*) L'hypothèse que la base du programme X est composée des m premiers vecteurs des conditions n'est faite que pour simplifier l'écriture et n'a naturellement aucune répercussion sur la généralité des raisonnements.

Tableau 7.3

Tableau principal de la l-ième itération

B	$c_{k_j j}$	$\widetilde{X}$	e_1	e_2	...	e_m	A_r	$\alpha(\beta)$	θ
A_1	$c_{k_1 1}$	x_1	e_{11}	e_{12}	...	e_{1m}	x_{1r}	$\alpha_1(\beta_1)$	θ_1
A_2	$c_{k_2 2}$	x_2	e_{21}	e_{22}	...	e_{2m}	x_{2r}	$\alpha_2(\beta_2)$	θ_2
.....	.....	.....	.....	.....	.....	.....	.....	.....	.....
A_k	$c_{k_k k}$	x_k	e_{k1}	e_{k2}	...	e_{km}	x_{kr}	$\alpha_k(\beta_k)$	θ_k
.....	.....	.....	.....	.....	.....	.....	.....	.....	.....
A_m	$c_{k_m m}$	x_m	e_{m1}	e_{m2}	...	e_{mm}	x_{mr}	$\alpha_m(\beta_m)$	θ_m
$m+1$	$F(X)$	x_r	λ_1	λ_2	...	λ_m	-1	$\alpha_r(\beta_r)$	θ_r

Dans la première colonne du tableau principal (cases 1 à m) figurent les coefficients angulaires des secteurs de linéarité des fonctions $f_j(x_j)$, $j = 1, 2, \ldots, m$, sur lesquels se situent les composantes correspondantes du programme X. Dans la colonne suivante (lignes $A_1, \ldots, A_m$) on inscrit les valeurs des composantes de base du programme X. Les m colonnes suivantes du tableau (lignes $A_1, \ldots, A_m$) sont prévues pour les éléments de la matrice

$$\| e_{ij} \| = (A_1, A_2, \ldots, A_m)^{-1}.$$

D'après la définition de la matrice inverse, les éléments de la j-ième colonne de la matrice $\| e_{ij} \|$ sont égaux aux coefficients correspondants de la décomposition du vecteur e_j par rapport aux vecteurs de la base $A_1, A_2, \ldots, A_m$. Seules ont été remplies jusqu'à présent les m premières lignes du tableau 7.3. Dans la dernière ($(m+1)$-ième) ligne de ce tableau figurent les composantes du vecteur $\Lambda = (\lambda_1, \lambda_2, \ldots, \lambda_m)$ et la valeur de la fonction à minimiser sur le programme en question (colonnes $e_1, \ldots, e_m$ et $c_{k_j j}$).

La formule (3.15) entraîne

$$\lambda_i = \sum_{s=1}^{m} c_{k_s s} e_{si}, \quad i = 1, 2, \ldots m. \tag{5.2}$$

Ainsi, pour obtenir λ_i, il suffit de multiplier les colonnes $e_{k_j j}$ et e_i du tableau principal (lignes $A_1, \ldots, A_m$).

Tout au cours de la procédure de résolution on construit ce qu'on appelle le tableau auxiliaire (tableau 7.4), dont six lignes sont remplies à chaque itération.

Tableau 7.4

Tableau auxiliaire

			A_1	A_2	...	A_m	A_{m+1}	...	A_{n-1}	A_n
	.....	.....	...	...	...	...	.....	...	.....	.....
	.....	.....	...	...	...	...	.....	...	.....	.....
	.....	.....	...	...	...	...	.....	...	.....	.....
Itération n° l	$6l-5$	k_j	—	—	...	—	k_{m+1}	...	k_{n-1}	k_n
	$6l-4$	z_j	—	—	...	—	z_{m+1}	...	z_{n-1}	z_n
	$6l-3$	c_{kjl}	—	—	...	—	$c_{k_{m+1}m+1}$	...	$c_{k_{n-1}n-1}$	$c_{k_n n}$
	$6l-2$	$\overline{\Delta}_{k_j j}$	—	—	...	—	$\overline{\Delta}_{k_{m+1}m+1}$	...	$\overline{\Delta}_{k_{n-1}n-1}$	$\overline{\Delta}_{k_n n}$
	$6l-1$	$c_{k_j-1,j}$	—	—	...	—	$c_{k_{m+1}-1,\, m+1}$	...	$c_{k_{n-1}-1,\, n-1}$	$c_{k_n-1,\, n}$
	$6l$	$\overline{\overline{\Delta}}_{k_j j}$	—	—	...	—	$\overline{\overline{\Delta}}_{k_{m+1}m+1}$	...	$\overline{\overline{\Delta}}_{k_{n-1}n-1}$	$\overline{\overline{\Delta}}_{k_n n}$

Dans le tableau auxiliaire considéré nous n'avons dégagé que les lignes qui sont remplies à la l-ième itération, c'est-à-dire les lignes aux numéros $(6l - 5)$ à $6l$.

Les cases de ces lignes qui correspondent aux vecteurs de la base restent vides. Dans la $(6l - 5)$-ième ligne du tableau 7.4 (ligne k_j) figurent les composantes du programme X associées aux vecteurs des conditions qui n'entrent pas dans la base. Dans le cas considéré il s'agit des composantes x_j avec $j \geqslant m + 1$.

Puisque ces composantes sont astreintes à coïncider avec certaines valeurs critiques ($x_j = d_{k_j j}$), il suffit d'indiquer les numéros k_j des valeurs critiques correspondantes.

Dans la ligne suivante (ligne z_j) on porte les valeurs

$$z_j = \sum_{i=1}^{m} a_{ij}\lambda_i, \quad j \geqslant m+1, \tag{5.3}$$

qui d'après la formule (5.3) s'obtiennent par multiplication de la $(m+1)$-ième ligne du tableau principal l (colonnes $e_1, \ldots, e_m$) et de la colonne A_j du tableau I.

Dans la ligne $6l-3$ on inscrit les nombres $c_{k_j j}$, $j = m+1, \ldots, n$. On trouve les valeurs $c_{k_j j}$ dans le tableau 7.2, d'après le numéro de la composante j et celui de la valeur critique $\gamma = k_j$. Constatons que si $k_j = l_j + 1$, la case correspondante de la ligne $6l-3$ reste vide.

Dans la ligne suivante (ligne $(\overline{\Delta}_{k_j j})$), on inscrit les quantités $\overline{\Delta}_{k_j j} = c_{k_j j} - z_j$, $j = m+1, \ldots, n$. Si une case quelconque de la ligne précédente est vide, elle n'est pas remplie non plus dans la ligne considérée.

Dans la ligne $6l-1$ on écrit les nombres $c_{k_j-1, j}$, $j = m+1, \ldots, n$, que l'on cherche encore dans le tableau 7.2 suivant les numéros j et $\gamma = k_j - 1$, la j-ième case de cette ligne n'étant remplie que si $k_j > 0$ et $\overline{\Delta}_{k_j j} \geqslant 0$.

Dans la ligne $6l$ du tableau auxiliaire on place les quantités $\overline{\overline{\Delta}}_{k_j j} = z_j - c_{k_j-1, j}$. Aux cases vides de la ligne $6l-1$ correspondent alors les cases également vides de la ligne $6l$.

Si tous les éléments des lignes $\overline{\Delta}_{k_j j}$ et $\overline{\overline{\Delta}}_{k_j j}$ sont non négatifs, le programme X est optimal et la procédure de résolution du problème prend fin. S'il n'en est pas ainsi, on poursuit le calcul des itérations.

Parmi les nombres $\overline{\Delta}_{k_j j}$ et $\overline{\overline{\Delta}}_{k_j j}$ qui figurent dans les lignes $6l-2$ et $6l$ du tableau auxiliaire, on choisit le plus petit que l'on désigne par $\Delta < 0$.

Deux cas sont alors possibles:

1°. $\Delta = \overline{\Delta}_{k_r r}$ avec un certain r.

2°. $\Delta = \overline{\overline{\Delta}}_{k_r r}$ avec un certain r *).

Dans les deux cas, on introduit dans la base le vecteur A_r. Considérons encore le tableau principal (7.3). Dans sa colonne A_r (lignes $A_1, \ldots, A_m$), on place les coefficients x_{ir} de décomposition du vecteur A_r par rapport aux vecteurs de la base du programme X.

---

*) Si parmi $\overline{\Delta}_{k_j j}$ et $\overline{\overline{\Delta}}_{k_j j}$ il y a plusieurs nombres qui sont les plus petits on choisit l'un quelconque d'entre eux.

Comme ces coefficients sont solution du système d'équations linéaires

$$A_X Z = A_r,$$

il vient

$$x_{ir} = \sum_{s=1}^{m} e_{is} a_{sr},$$

ce qui signifie que le i-ème coefficient est formé par multiplication de la ligne A_i du tableau principal (colonnes $e_1, \ldots, e_m$) et de la colonne A_r du tableau 7.1.

Dans la dernière ($(m+1)$-ième) case de la colonne A_r du tableau principal, on inscrit la valeur $x_{rr} = -1$.

Dans la dernière case de la colonne $\widetilde{X}$, on fait figurer la valeur de la composante affectée au vecteur à ajouter $x_r = d_{k_r r}$. On trouve cette valeur critique dans le tableau 7.2 d'après les numéros $j = r$ et $\gamma = k_r$. Les deux dernières colonnes du tableau principal 7.3 servent pour déterminer le nombre θ'.

Dans le cas 1°, on porte dans la j-ième case de la colonne α (β) les nombres

$$\alpha_j \quad \text{si} \quad x_{jr} > 0,$$
$$\beta_j \quad \text{si} \quad x_{jr} < 0.$$

Dans le cas 2°, cette case est occupée par les valeurs

$$\alpha_j \quad \text{si} \quad x_{jr} < 0,$$
$$\beta_j \quad \text{si} \quad x_{jr} > 0.$$

Si $x_{jr} = 0$, ou s'il faut porter $\alpha_j = -\infty$ ($\beta_j = \infty$) dans la j-ième case de la colonne α (β), cette case reste vide. On range dans la colonne θ du tableau principal les quantités

$$\theta_j = \begin{cases} \dfrac{x_j - \alpha_j(\beta_j)}{x_{jr}} & \text{pour le cas 1°,} \\[2ex] \dfrac{\alpha_j(\beta_j) - x_j}{x_{jr}} & \text{pour le cas 2°.} \end{cases} \tag{5.4}$$

Notons qu'aux cases vides de la colonne α (β) correspondent les cases vides de la colonne θ. Parmi les nombres θ_j on choisit le plus petit, qui est posé égal à θ'. Pour définir le nombre θ'', on construit le tableau supplémentaire de la l-ième itération (tableau 7.5). Ce tableau est composé pour certains des vecteurs $A_1, A_2, \ldots, A_m, A_r$.

Pour qu'un vecteur A_j soit porté sur le tableau supplémentaire, il faut que:

a) la j-ième case de la colonne θ du tableau principal corresponde à $x_{jr} \neq 0$;

b) qu'il y ait entre les nombres qui figurent dans la j-ième case des colonnes $\widetilde{X}$ et α (β) au moins une valeur critique de l'argument

x_j (dans ces conditions, il faut tenir compte de l'éventualité $\alpha = -\infty$ et $\beta = \infty$).

Tableau 7.5

Tableau supplémentaire de la l-ième itération

	.....	A_j				.....
$\theta_{\gamma j}$	.....					.....
γ	.....					.....
N	.....					.....
δ_l	.....					.....
Δ_l	.....					.....

Supposons qu'un vecteur A_j doive être pris en compte dans le tableau supplémentaire. C'est un vecteur auquel correspondent, d'une façon générale, plusieurs sous-colonnes du tableau. Le nombre de sous-colonnes associées au vecteur A_j est égal à celui des quantités

$$\theta_{\gamma j} = \begin{cases} \dfrac{x_j - d_{\gamma j}}{x_{jr}} & \text{pour le cas } 1^\circ, \\ \dfrac{d_{\gamma j} - x_j}{x_{jr}} & \text{pour le cas } 2^\circ. \end{cases} \tag{5.5}$$

Les valeurs critiques $d_{\gamma j}$ sont comprises ici entre les nombres *) figurant dans la j-ième case des colonnes $\tilde{X}$ et α (β) du tableau principal l; elles sont telles que $\theta_{\gamma j} \leqslant \theta'$. Dans la première ligne du tableau supplémentaire (dans les sous-colonnes correspondant au vecteur A_j) on inscrit les quantités $\theta_{\gamma j}$, calculées d'après la formule (5.5). Dans la ligne suivante (ligne γ), on écrit les numéros des valeurs critiques $d_{\gamma j}$ pour lesquelles on a calculé les valeurs $\theta_{\gamma j}$ correspondantes.

En examinant l'un après l'autre les vecteurs de la base et le vecteur à introduire, on remplit les lignes $\theta_{\gamma j}$ et γ pour ceux d'entre eux qu'il faut inclure dans le tableau supplémentaire. On numérote ensuite les quantités situées dans la ligne $\theta_{\gamma j}$ (dans les limites de toutes les colonnes du tableau) dans l'ordre de leur croissance et l'on place les numéros correspondants dans la ligne N. On forme de ce fait la suite (3.22). Notons que dans le cas dégénéré la ligne $\theta_{\gamma j}$ peut comporter des éléments analogues quant à leur valeur; l'ordre de numérotation, sur l'ensemble de ces éléments, peut être arbitraire.

---

*) Si x_j ($j = 1, 2, \ldots, m$) est égal à l'une des valeurs critiques $d_{\gamma j}$, cette composante est alors considérée comme située à l'intérieur du segment $[d_{kjj}, d_{kj+1,\, j}]$ défini par le nombre c_{kjj}.

Les deux lignes suivantes du tableau supplémentaire sont remplies simultanément; l'ordre suivi pour les remplir est déterminé par les numéros disposés dans la ligne N.

Chaque case d'une ligne quelconque du tableau supplémentaire 7.5 est déterminée par les numéros j et γ. Supposons qu'après avoir composé la ligne N, on ait obtenu la suite de cases

$$(\gamma_1, j_1), \quad (\gamma_2, j_2), \ \ldots, \ (\gamma_s, j_s).$$

On écrit dans la ligne δ_t les quantités

$$\delta_t = \sigma_{\gamma_t j_t} \, | x_{j_t r} |, \quad t = 1, 2, \ldots, s.$$

On met dans la ligne Δ_t les nombres

$$\Delta_t = \Delta + \sum_{i=1}^{t} \delta_i = \Delta_{t-1} + \delta_t, \quad t = 1, 2, \ldots, s.$$

Remarquons que les quantités δ_t et Δ_t figurent dans la case qui a dans la ligne N le numéro t. La composition des lignes δ_t et Δ_t se poursuit soit tant qu'on obtienne le premier élément non négatif de la ligne Δ_t, soit tant que toutes les cases soient occupées. Dans ce dernier cas, $\Delta_t < 0$ pour $t = 1, 2, \ldots, s$.

Dans le premier cas, $\theta'' = \theta_{\gamma_\nu j_\nu}$, où ν est le numéro du premier élément non négatif de la ligne Δ_t égal à Δ_ν. Dans le deuxième cas, θ'' est posé égal à ∞. Ainsi, au terme de la construction du tableau supplémentaire, nous déterminons θ'', et aussi les vecteurs des conditions sur lesquels cette grandeur est atteinte. Définissons le nombre θ_0 comme le plus petit des nombres θ' et θ''.

Si $\theta_0 = \min(\theta', \theta'') = \infty$ (c'est-à-dire si $\theta' = \theta'' = \infty$), le problème n'a pas de solution. Supposons que $\theta_0 < \infty$. Le vecteur à retrancher de la base est choisi parmi les vecteurs A_j, $j = 1, 2, \ldots, m, r$, tels que soit $\theta_{\gamma j} = \theta'' = \theta_0$ avec un certain γ (ces vecteurs sont sélectionnés au cours de la construction du tableau supplémentaire), soit $\theta_{\gamma j} = \theta' = \theta_0$ (ces vecteurs sont définis en composant la dernière colonne du tableau principal).

Désignons par E l'ensemble des indices des vecteurs A_j indiqués ci-dessus. Nous avons déjà dit à plus d'une reprise que dans le cas non dégénéré l'ensemble E se compose d'un seul élément, et le vecteur à éliminer de la base est défini univoquement. Si E se compose de plusieurs éléments, on peut appliquer la règle décrite au § 4, avec les vecteurs $R_1, R_2, \ldots, R_m$ définis par la formule (5.1).

Marquons (par le signe —, par exemple) les colonnes e_j du tableau principal 7.3 pour lesquelles, d'après la formule (5.1), $R_j = -e_j$ (e_j est ici le vecteur unité).

On examine les rapports $\frac{e_{i1}}{x_{ir}}$ avec $i \in E$ (convenons de considérer $e_{rs} = 0$ pour tout s). Désignons par E' l'ensemble des indices $i \in E$

pour lesquels ces rapports atteignent leur minimum (leur maximum). On choisit le minimum soit si l'on a le cas 1° et si la colonne e_1 n'est pas marquée, soit si l'on a le cas 2° et si la colonne e_1 est marquée. Mais si la colonne e_1 est ou n'est pas marquée, suivant que l'on a le cas 1° ou le cas 2°, on cherche alors le maximum. Si l'ensemble E' compte un seul élément, le numéro du vecteur à exclure est déjà trouvé. Dans le cas contraire, on passe à la colonne e_2 du tableau principal et, de façon analogue, on dégage de l'ensemble des indices E' un certain sous-ensemble E'' de cet ensemble. La procédure se poursuit jusqu'à l'obtention d'un ensemble comportant un seul élément qui définit le vecteur à retrancher de la base.

Constatons que dans les problèmes pratiques la probabilité du cyclage est très faible. C'est pourquoi, au lieu de la règle décrite, on peut se servir d'une règle plus simple: on élimine de la base n'importe lequel des vecteurs A_j avec $j \in E$.

Supposons que le vecteur à éliminer de la base soit déjà trouvé; posons qu'il s'agisse du vecteur A_k. Construisons un nouveau tableau principal en considérant que $k \neq r$. Les éléments correspondants des deux tableaux ne se distingueront que par une apostrophe. Les quantités e'_{ij}, $i = 1, 2, \ldots, m$, sont les coefficients de décomposition du vecteur e_j par rapport à la nouvelle base, formée à partir de l'ancienne en remplaçant A_k par A_r. C'est pourquoi, pour le calcul des e_{ij}, on peut se servir de formules de récurrence connues (cf. par exemple, [87], p. 357). Ainsi,

$$e'_{ij} = \begin{cases} e_{ij} - e_{kj}\dfrac{x_{ir}}{x_{kr}}, & i \neq k, \\ e_{kj}\dfrac{1}{x_{kr}}, & i = k. \end{cases} \tag{5.6}$$

Il peut arriver que $r = k$, c'est-à-dire que la nouvelle base se confonde avec l'ancienne. Dans ce cas, il est clair que $e'_{ij} = e_{ij}$. La colonne $\widetilde{X}$ du tableau principal $l + 1$ est composée de valeurs x'_j pour $j = 1, 2, \ldots, k-1, r, k+1, \ldots, m$, définies d'après les formules (3.19) ou (3.26) suivant que c'est le cas 1° ou 2° qui a lieu. La ligne $c_{k_j j}$ du nouveau tableau principal est composée à l'aide de la colonne $\widetilde{X}$ et du tableau 7.2. Si x'_j se situe à l'intérieur d'un segment de linéarité de $f_j(x)$, on met alors dans la case correspondante de la ligne $c_{k_j j}$ le coefficient angulaire de $f_j(x)$ sur ce segment. Dans le cas où x'_j ne coïncide avec aucune des valeurs critiques du j-ième argument, le segment de linéarité considéré est défini univoquement. Si, par contre, $x'_j = d_{\gamma j}$ avec un certain γ, le choix du segment de linéarité de la fonction $f_j(x)$ qui contient x'_j peut se faire d'après la règle suivante.

a) Si le vecteur A_j appartenait à la base précédente, le segment de linéarité de la fonction $f_j(x)$ qui contient x_j' se situe du même côté de $d_{\nu j} = x_j'$ que le segment qui contient x_j.

b) Si $A_j = A_r$ est le vecteur introduit à cette étape, x_j appartient à $[d_{\nu-1,\, j},\ d_{\nu j}]$ ou à $[d_{\nu j},\ d_{\nu+1,\, j}]$ suivant que l'on a le cas 1° ou 2°. Remarquons que cette règle correspond à la recommandation donnée ci-dessus relative à la recherche du vecteur à éliminer de la base et qui rend le cyclage parfaitement impossible.

La case du coin gauche inférieur du nouveau tableau principal est prévue pour $F(X')$ qui est la valeur de la fonction à minimiser sur le programme X'. Cette valeur peut être calculée par substitution directe ou bien d'après la formule de récurrence

$$F(X') = F(X) + \Delta\theta_0 + \sum_{i=1}^{\nu} \sigma_{\nu_i j_i} |x_{j_i r}| (\theta_0 - \theta_{\nu_i j_i}) =$$

$$= F(X) + \Delta\theta_0 + \sum_{i=1}^{\nu} \delta_i (\theta_0 - \theta_{\nu_i j_i}) = F(X) + \theta_0 \Delta_\nu - \sum_{i=1}^{\nu} \delta_i \theta_{\nu_i j_i}, \quad (5.7)$$

où ν est le plus grand des i tels que

$$\theta_{\nu_i j_i} < \theta_0.$$

Les autres cases du tableau principal $l + 1$ sont remplies d'après les règles décrites auparavant. Sur ce prend fin l'itération l. Ainsi, chacune des itérations de l'algorithme consiste à construire les tableaux principal et supplémentaire et à prolonger de six lignes le tableau auxiliaire. La procédure de résolution s'achève par la recherche du programme optimal ou par la constatation de l'irrésolubilité du problème. Dans le premier cas, les composantes du programme optimal sont fournies par la colonne $\tilde{X}$ du dernier tableau principal et par la ligne k_j du tableau auxiliaire qui correspond à la dernière itération.

5.3. L'algorithme décrit peut également être utilisé pour résoudre les problèmes de maximisation de la fonction étudiée $F(X)$. Les fonctions $f_j(x_j)$ doivent être alors convexes vers le haut, c'est-à-dire pour des i et j quelconques

$$c_{ij} > c_{i+1,\, j}.$$

Il est clair qu'un tel problème se ramène au problème de minimisation de la fonction $\overline{F}(X) = \sum_{j=1}^{n} \overline{f}_j(x_j)$, où

$$\overline{f}_j(x_j) = -f_j(x_j) \qquad (j = 1, 2, \ldots, n).$$

On en déduit que pour résoudre des problèmes de maximisation il ne faut qu'apporter dans l'algorithme les modifications suivantes:

1. Par définition,

$$\sigma_{ij} = c_{i-1,j} - c_{ij}.$$

2. Les conditions d'optimalité du programme sont définies par la *non-positivité* des nombres $\overline{\Delta}_{kjj}$, $\overline{\overline{\Delta}}_{kjj}$ ($\overline{\overline{\Delta}}_{kjj}$ ne doit être calculé que pour $\overline{\Delta}_{kjj} \leqslant 0$).

3. Le vecteur A_r à ajouter à la nouvelle base est choisi tel qu'il assure le *maximum* des nombres $\overline{\Delta}_{kjj}$ et $\overline{\overline{\Delta}}_{kjj}$.

4. Pour le cas considéré, la formule de récurrence (5.7) se met sous la forme

$$F(X') = F(X) + \theta_0 \Delta_\nu + \sum_{i=1}^{\nu} \delta_i \theta_{\gamma_i j_i},$$

où ν est, comme auparavant, le plus grand des i tel que

$$\theta_{\gamma_i j_i} < \theta_0.$$

5. Le calcul des quantités Δ_t se fait d'après la formule

$$\Delta_t = \Delta - \sum_{i=1}^{t} \delta_i = \Delta_{t-1} - \delta_t.$$

Le calcul des nombres Δ_t se poursuit tant qu'on obtienne le premier nombre *non positif* Δ_ν, ou bien tant que toutes les cases de la ligne Δ_t soient remplies. Dans le premier cas $\theta'' = \theta_{\gamma_\nu j_\nu}$, dans le deuxième $\theta'' = \infty$.

§ 6. Exemple

Illustrons l'algorithme de résolution du problème par un exemple numérique. Pour pouvoir apprécier l'avantage du passage d'un problème particulier de programmation linéaire à un problème linéaire par morceaux, nous commencerons par énoncer le problème linéaire.

Soit à maximiser la forme linéaire

$$\left.\begin{aligned} L(X) = {} & 8x_1 + 9x_2 + 8x_3 + 6x_4 + 8x_5 + 9x_6 + 8x_7 + 11x_8 - \\ & - \{3x_1^{(1)} + 2x_2^{(1)} + 4x_3^{(1)} + 3x_5^{(1)} + x_6^{(1)} + x_7^{(1)} + x_8^{(1)} + \\ & + x_1^{(2)} + x_2^{(2)} + 3x_3^{(2)} + 2x_7^{(2)} + 2x_8^{(2)} + \\ & + 2x_1^{(3)} + x_2^{(3)} + 3x_8^{(3)} + \\ & + 2x_2^{(4)} + x_8^{(4)}\} \end{aligned}\right\} \quad (6.1)$$

sous les conditions

$$\left.\begin{aligned} x_1 - 2x_2 + 3x_3 - x_4 + 5x_5 - 3x_6 + x_7 - x_8 &\leqslant 10, \\ 3x_1 + x_2 - x_3 + 2x_4 + 2x_5 - 2x_6 - 3x_7 + 4x_8 &\leqslant 8, \\ -2x_1 + x_2 + x_3 + x_5 + x_6 - 2x_8 &\leqslant 7, \\ 5x_1 - 4x_2 + 3x_4 - 2x_5 - x_7 &\leqslant 5, \end{aligned}\right\} \quad (6.2)$$

$$\left.\begin{array}{llll}
0 \leqslant x_1 \leqslant 4; & 0 \leqslant x_2 \leqslant 5; & 0 \leqslant x_3 \leqslant 6; & 0 \leqslant x_4; \\
x_1 \leqslant 1 + x_1^{(1)}; & x_2 \leqslant 1 + x_2^{(1)}; & x_3 \leqslant 2 + x_3^{(1)}; & \\
x_1 \leqslant 2 + x_1^{(2)}; & x_2 \leqslant 2 + x_2^{(2)}; & x_3 \leqslant 5 + x_3^{(2)}; & \\
x_1 \leqslant 3 + x_1^{(3)}; & x_2 \leqslant 3 + x_2^{(3)}; & & \\
 & x_2 \leqslant 4 + x_2^{(4)}; & & \\
0 \leqslant x_5 \leqslant 2; & 0 \leqslant x_6; & 0 \leqslant x_7 \leqslant 7; & 0 \leqslant x_8 \leqslant 5; \\
x_5 \leqslant 1 + x_5^{(1)}; & x_6 \leqslant 2 + x_6^{(1)}; & x_7 \leqslant 2 + x_7^{(1)}; & x_8 \leqslant 1 + x_8^{(1)}; \\
 & & x_7 \leqslant 4 + x_7^{(2)}; & x_8 \leqslant 2 + x_8^{(2)}; \\
 & & & x_8 \leqslant 3 + x_8^{(3)}; \\
 & & & x_8 \leqslant 4 + x_8^{(4)};
\end{array}\right\} \quad (6.3)$$

$$x_j^{(i)} \geqslant 0. \qquad (6.4)$$

Le problème linéaire par morceaux équivalent (problème (B)) consiste, conformément aux raisonnements du paragraphe 1.3, à maximiser la forme linéaire par morceaux

$$F(X) = \sum_{j=1}^{n} f_j(x_j) \qquad (6.5)$$

sous les conditions (6.2) et

$$\left.\begin{array}{llll}
0 \leqslant x_1 \leqslant 4; & 0 \leqslant x_2 \leqslant 5; & 0 \leqslant x_3 \leqslant 6; & x_4 \geqslant 0; \\
0 \leqslant x_5 \leqslant 2; & x_6 \geqslant 0; & 0 \leqslant x_7 \leqslant 7; & 0 \leqslant x_8 \leqslant 5.
\end{array}\right\} \quad (6.6)$$

Les fonctions $f_j(x_j)$ se calculent d'après les formules (1.38)

$$f_1(x_1) = \begin{cases} 8x_1 & 0 \leqslant x_1 \leqslant 1, \\ 5x_1 + 3 & 1 \leqslant x_1 \leqslant 2, \\ 4x_1 + 5 & 2 \leqslant x_1 \leqslant 3, \\ 2x_1 + 11 & 3 \leqslant x_1 \leqslant 4. \end{cases}$$

$$f_2(x_2) = \begin{cases} 9x_2 & 0 \leqslant x_2 \leqslant 1, \\ 7x_2 + 2 & 1 \leqslant x_2 \leqslant 2, \\ 6x_2 + 4 & 2 \leqslant x_2 \leqslant 3, \\ 5x_2 + 7 & 3 \leqslant x_2 \leqslant 4, \\ 3x_2 + 15 & 4 \leqslant x_2 \leqslant 5. \end{cases}$$

$$f_3(x_3) = \begin{cases} 8x_3 & 0 \leqslant x_3 \leqslant 2, \\ 4x_3 + 8 & 2 \leqslant x_3 \leqslant 5, \\ x_3 + 23 & 5 \leqslant x_3 \leqslant 6. \end{cases}$$

$$f_4(x_4) = 6x_4 \qquad x_4 \geqslant 0.$$

$$f_5(x_5) = \begin{cases} 8x_5 & 0 \leqslant x_5 \leqslant 1, \\ 5x_5 + 3 & 1 \leqslant x_5 \leqslant 2. \end{cases}$$

$$f_6(x_6) = \begin{cases} 9x_6 & 0 \leqslant x_6 \leqslant 2, \\ 8x_6 + 2 & x_6 \geqslant 2. \end{cases}$$

$$f_7(x_7) = \begin{cases} 8x_7 & 0 \leqslant x_7 \leqslant 2, \\ 7x_7 + 2 & 2 \leqslant x_7 \leqslant 4, \\ 5x_7 + 10 & 4 \leqslant x_7 \leqslant 7. \end{cases}$$

$$f_8(x_8) = \begin{cases} 11x_8 & 0 \leqslant x_8 \leqslant 1, \\ 10x_8 + 1 & 1 \leqslant x_8 \leqslant 2, \\ 8x_8 + 5 & 2 \leqslant x_8 \leqslant 3, \\ 5x_8 + 14 & 3 \leqslant x_8 \leqslant 4, \\ 4x_8 + 18 & 4 \leqslant x_8 \leqslant 5. \end{cases}$$

Tableau 7.6

I

i \ A_j	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}	A_{11}	A_{12}	B
1	1	−2	3	−1	5	−3	1	−1	1				10
2	3	1	−1	2	2	−2	−3	4		1			8
3	−2	1	1		1	1		−2			1		7
4	5	−4		3	−2		−1					1	5

Tableau 7.7

II

γ \ j	1			2			3			4			5			6			7			8		
	d	σ	c	d	σ	c	d	σ	c	d	σ	c	d	σ	c	d	σ	c	d	σ	c	d	σ	c
0	0		8	0		9	0		8	0		6	0		8	0		9	0		8	0		11
1	1	3	5	1	2	7	2	4	4	∞			1	3	5	2	1	8	2	1	7	1	1	10
2	2	1	4	2	1	6	5	3	1				2			∞			4	2	5	2	2	8
3	3	2	2	3	1	5	6												7			3	3	5
4	4			4	2	3																4	1	4
5				5																		5		

Constatons que dans le § 5 le schéma de calcul est décrit comme étant appliqué au problème de minimisation d'une fonction linéaire par morceaux. Pour une meilleure maîtrise des modifications à apporter dans l'algorithme indiquées au paragraphe 5.3 et liées à la recherche d'un maximum, nous considérons ici un exemple de maximisation de la fonction (6.5) sous les conditions (6.2), (6.6).

Remplaçons les contraintes (6.2), mises sous la forme d'inégalités, par des égalités. A cet effet introduisons les variables additionnelles non négatives x_9, x_{10}, x_{11} et x_{12}

$$x_9 \geqslant 0; \quad x_{10} \geqslant 0; \quad x_{11} \geqslant 0; \quad x_{12} \geqslant 0.$$

Dressons le tableau I (tableau 7.6) qui contient les vecteurs des conditions A_j du problème B, et le tableau II (tableau 7.7), dans lequel sont écrites les valeurs des points critiques d_{ij} et des coefficients σ_{ij} et $c_{ij} = c_{i-1,\,j} - \sigma_{ij}$. Les variables additionnelles x_9, x_{10}, x_{11}, x_{12} ont chacune un point critique $d_{i0} = \alpha_i = 0$, et les valeurs c_{ij} et σ_{ij} qui leur correspondent sont égales à zéro.

Convenons de laisser vides les cases des tableaux qui doivent être remplies par des éléments nuls et de tracer des traits dans les cases qui ne doivent pas être remplies dans la procédure de résolution.

Il est naturel de prendre pour base initiale le système de vecteurs unités qui corresponde aux variables additionnelles.

Composons le tableau principal 1 (tableau 7.8). La colonne $\widetilde{X}^{(1)}$ (ou plus exactement les $m = 4$ premières cases de cette colonne) contient les composantes

Tableau 7.8

Tableau principal 1

B	c_{kjj}	$\widetilde{X}^{(1)}$	e_1	e_2	e_3	e_4	A_9	α (β)	θ
A_9		10	1				—1	—	—
A_{10}		8		1			4		2
A_{11}		7			1		—2	—	—
A_{12}		5				1		—	—
$m+1$							—1	5	5

de base du programme initial. Dans notre cas, ce sont les composantes du vecteur des contraintes. La colonne c_{kjj} se compose des coefficients angulaires de la fonction $f_j(x)$ sur ses intervalles de linéarité. Pour les variables additionnelles, $c_{kjj} = 0$. Les colonnes $e_1, \ldots, e_4$ (leurs m premières cases) contiennent les coefficients de décomposition des vecteurs unités par rapport aux vecteurs de la base. Dans notre cas, les vecteurs de la base correspondent aux variables additionnelles et les colonnes $e_1, \ldots, e_4$ forment une matrice unité.

La dernière $(m + 1)$-ième ligne du tableau principal contient les valeurs λ_i calculées d'après la formule (5.2). Dans notre cas, tout $\lambda_i = 0$.

Il est clair que pour le programme initial $X^{(1)}$ choisi la quantité $F(X^{(1)}) = 0$. La valeur de $F(X^{(1)})$ est inscrite dans la dernière case de la colonne c_{kjj}.

Commençons par composer le tableau auxiliaire 7.9. Pour les vecteurs de la base (à la 1-ère itération, pour A_9, A_{10}, A_{11}, A_{12}), le tableau n'est pas rempli.

Tableau 7.9

Tableau auxiliaire (1-ère itération)

	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}	A_{11}	A_{12}
k_j									—	—	—	—
z_j									—	—	—	—
c_{kjj}	8	9	8	6	8	9	8	11	—	—	—	—
$\bar{\Delta}_{kjj}$	8	9	8	6	8	9	8	$\boxed{11}$	—	—	—	—
$c_{kj-1,j}$	—	—	—	—	—	—	—	—	—	—	—	—
$\bar{\bar{\Delta}}_{kjj}$	—	—	—	—	—	—	—	—	—	—	—	—

Sur la première ligne du tableau auxiliaire on porte les numéros k_j des valeurs critiques avec lesquelles coïncident les composantes secondaires du programme $x_j^{(1)}$ avec $j = 1, 2, \ldots, 8$. A ces composantes répondent les vecteurs des conditions non inclus dans la base. Dans notre cas, toutes les variables principales sont égales à zéro et correspondent par conséquent à des valeurs critiques nulles. C'est pourquoi la première ligne du tableau n'est pas remplie. On ne remplit pas non plus la deuxième ligne qui contient les valeurs de z_j. A la première itération, tout $\lambda_i = 0$. Par conséquent, $z_j = \sum_{i=1}^{k} a_{ij}\lambda_i = 0$.

Dans la troisième ligne figurent les valeurs de c_{kjj}. Dans notre cas, $c_{kjj} = c_{0j} = c_j$.

La ligne suivante contient les valeurs de $\bar{\Delta}_{kjj} = c_{kjj} - z_j$. Elle est composée suivant les éléments correspondants des deux lignes précédentes. Sur la cinquième ligne on porte les valeurs de $c_{kj-1,j}$ nécessaires au calcul de $\bar{\bar{\Delta}}_{kjj} = z - c_{kj-1,j}$. Dans la dernière ligne se trouvent les valeurs de $\bar{\bar{\Delta}}_{kjj}$. Les quantités $\bar{\bar{\Delta}}_{kjj}$ et, par conséquent, les coefficients angulaires $c_{kj-1,j}$ ne se calculent pas avec $k_j = 0$ non plus dans les cas où $\bar{\Delta}_{kjj} > 0$. C'est précisément le cas de la première étape des calculs.

Conformément au critère d'optimalité, on conclut que le programme initial n'est pas optimal. Il faut passer à une nouvelle base, pour laquelle on peut s'attendre que la valeur de F augmente. Nous sommes en présence ici du cas 1°.

La plus grande valeur de $\overline{\Delta}_{kjj}$ (encadrée dans le tableau auxiliaire 7.9) est $\overline{\Delta}_{08} = 11$, elle correspond au vecteur A_8. Donc, $r = 8$, et il faut tester s'il est rationnel d'introduire dans la base le vecteur A_8. Les coefficients de décomposition x_{jr} du vecteur A_8 par rapport aux vecteurs de la base sont portés sur la colonne A_8 du tableau principal 1. La base initiale se composant de vecteurs unités, on a $x_{jr} = x_{j8} = a_{j8}$.

La dernière $(m + 1)$-ième case de la colonne A_r contient toujours le nombre $x_{rr} = -1$. Dans la $(m + 1)$-ième case de la colonne $\widetilde{X}^{(1)}$ figure la valeur de x_r. La quantité $x_r = d_{k_r r}$ est déterminée à partir du tableau II (tableau 7.7) d'après le numéro k_r qui figure dans la première ligne du tableau auxiliaire. Dans notre cas, $k_r = 0$ et $x_r = 0$.

Passons à la définition du vecteur qu'il faut éliminer de la base. A cette fin il faut calculer θ' et θ'' et trouver le vecteur de la base qui correspond à la grandeur $\theta_0 = \min(\theta', \theta'')$.

Commençons par calculer θ'. Composons la colonne α (β) qui, dans le tableau principal, vient après la colonne A_r. Rappelons qu'on porte dans la colonne α (β) les valeurs α_j (borne gauche de l'intervalle de définition de x_j) si x_{jr} (élément correspondant de la colonne A_r) est positif, et β_j (borne droite de l'intervalle de définition de x_j) si $x_{jr} < 0$.

Si l'intervalle de définition de x_j n'est borné ni à gauche ni à droite, ou si $x_{jr} = 0$, la case correspondante de la colonne α (β) n'est pas remplie et n'est pas prise en compte lors du calcul de θ_j.

D'après les remarques qui ont été faites, on ne remplit dans le tableau principal 1 (tableau 7.8) que deux cases de la colonne α (β) : $\alpha_{10} = 0$ et $\beta_8 = \beta_r = 5$. (Les valeurs $\alpha_j = d_{0j}$ et $\beta_j = d_{l_{j+1}, j}$, points critiques extrémaux associés aux diverses variables, sont inscrites dans le tableau II). Les cases de la colonne α (β) marquées d'un trait correspondent aux valeurs $\beta_9 = \beta_{11} = \infty$ et $x_{12,8} = 0$.

Pour calculer les éléments de la dernière colonne θ_j, il faut soustraire des éléments de la colonne $\widetilde{X}^{(1)}$ les valeurs correspondantes de la colonne α (β) et diviser par les composantes de la colonne A_8. Ceci concerne la totalité des $m + 1$ cases des colonnes signalées. Les cases de la colonne θ qui correspondent aux cases cochées de la colonne α (β) ne sont pas remplies. Ainsi, dans la colonne θ, on remplit seulement les deux cases

$$\theta_{10} = \frac{x_{10} - 0}{x_{10,8}} = \frac{8}{4} = 2 \quad \text{et} \quad \theta_r = \theta_8 = \frac{0 - 5}{-1} = 5.$$

Le plus petit des éléments θ_j est égal à θ'. Dans notre cas, $\theta' = \theta_{10} = 2$.

Passons à la détermination de θ''. Son calcul se fait en trois étapes. On compose d'abord la suite non décroissante $\theta_{\nu j}$, puis on calcule les quantités Δ_t, et enfin, on définit directement θ''. Tous les calculs relatifs à la recherche de θ'' sont portés sur le tableau supplémentaire 7.10.

Rappelons que les colonnes du tableau supplémentaire sont déterminées par les vecteurs de la base et par le vecteur A_r qu'on se propose d'introduire dans la base. Le tableau est dressé progressivement, puisque le nombre de sous-colonnes en lesquelles doit être divisée chaque colonne s'établit au cours des calculs.

La colonne associée au vecteur A_j n'est incluse dans le tableau supplémentaire que si l'une des deux conditions suivantes est remplie:

a) $x_{jr} > 0$ et il y a à gauche de x_j des valeurs critiques de l'argument x_j non égales à α_j,

b) $x_{jr} < 0$ et il y a à droite de x_j des valeurs critiques de l'argument x_j non égales à β_j.

Si, notamment, $x_{jr} = 0$, la colonne correspondant à A_j n'est pas incluse dans le tableau supplémentaire.

Tableau 7.10

Tableau supplémentaire 1

	A_8	
$\theta_{\gamma j}$	1	2
γ	1	2
N	1	2
σ_t	1	2
Δ_t	10	8

Dans notre cas, on n'inclut pas dans le tableau les colonnes correspondant aux vecteurs A_9 ($x_{jr} = x_{98} = -1 < 0$, mais x_9 est une variable non bornée supérieurement), A_{10} ($x_{10,8} = 4 > 0$, mais à gauche de $x_{10} = 8$ il n'y a qu'un point critique $\alpha_{10} = 0$), A_{11} ($x_{11,8} = -2 < 0$, mais x_{11} est non bornée) et A_{12} ($x_{12,8} = 0$). Par conséquent, le premier tableau supplémentaire se compose de la seule colonne $A_r = A_8$ ($x_{88} = -1 < 0$). C'est pourquoi en calculant $\theta_{\gamma j} = \theta_{\gamma 8}$, il faut tenir compte des points critiques à droite de $x_8 = 0$, c'est-à-dire de $d_{18} = 1$, $d_{28} = 2$, $d_{38} = 3$, $d_{48} = 4$ ($d_{58} = \beta_8$ est le point critique extrémal à droite de la variable x_8, qui n'est pas pris en considération et pour ce point θ_{58} n'est pas calculé).

Les nombres $\theta_{\gamma j}$ se calculent d'après la formule

$$\theta_{\gamma j} = \frac{x_j - d_{\gamma j}}{x_{jr}},$$

où γ sont les numéros des points critiques.

Nous savons que les valeurs $\theta_{\gamma j} > \theta'$ ne se calculent pas et ne sont pas portées dans le tableau supplémentaire. Dans notre cas, $\theta_{18} = 1$, $\theta_{28} = 2$. Comme $\theta' = 2$, il n'y a déjà plus de raison de calculer les valeurs de θ_{38} et θ_{48}, qui sont supérieures à θ_{28}.

Par conséquent, l'unique colonne A_8 du tableau supplémentaire 1 (tableau 7.10) se compose de deux sous-colonnes correspondant aux numéros $\gamma = 1$ et $\gamma = 2$ des points critiques.

Dans la première ligne du tableau figurent les valeurs de $\theta_{\gamma j}$, dans la deuxième, les numéros γ. La troisième ligne contient les numéros d'ordre des $\theta_{\gamma j}$ (dans l'ordre de croissance de $\theta_{\gamma j}$). La numérotation unifiée des $\theta_{\gamma j}$ définit l'ordre de succession dans lequel est calculé Δ_t. Cette numérotation unifiée est indispensable si le tableau supplémentaire contient plusieurs colonnes A_j.

Dans notre cas, $\gamma = 1$ et 2 et $N = 1$ et 2. Dans la quatrième ligne du tableau supplémentaire on inscrit les valeurs $\delta_t = \delta_{\gamma_t j_t} \mid x_{\gamma_t j}$, nécessaires pour

calculer Δ_t. Dans la dernière ligne les valeurs de Δ_t:

$$\Delta_t = \Delta - \sum_{i=1}^{t} \sigma_{\gamma_i j_i} | x_{\gamma_i j} | = \Delta_{t-1} - \sigma_{\gamma_t j_t} | x_{j_t r} | = \Delta_{t-1} - \delta_t.$$

Dans notre cas,

$$\delta_1 = \sigma_{18} | x_{88} | = 1 \cdot | -1 | = 1, \ \delta_2 = \sigma_{28} | x_{88} | = 2,$$
$$\Delta_1 = \Delta_0 - \delta_1 = 11 - 1 = 10, \quad \Delta_2 = \Delta_1 - \delta_2 = 10 - 2 = 8.$$

$\Delta_0 = \Delta = \overline{\Delta}_8 = 11$ est la plus grande valeur de $\overline{\Delta}_j$ tirée du tableau auxiliaire.

Les lignes δ_t et Δ_t sont remplies simultanément jusqu'à ce que la valeur Δ_t devienne non positive ou jusqu'à ce que toutes les cases de la ligne Δ_t soient occupées. Dans ce dernier cas (qui a justement lieu dans le tableau supplémentaire 7.10), on pose $\theta'' = \infty$. (Rappelons que si nous avions obtenu un Δ_t négatif, il aurait fallu prendre pour θ'' la valeur de $\theta_{\gamma j}$ associée au premier Δ_t non positif.)

Ainsi, $\theta' = 2$, $\theta'' = \infty$.

$$\theta_0 = \min(\theta', \theta'') = \theta' = 2.$$

La valeur $\theta_0 = 2$ correspond à deux vecteurs, le vecteur A_{10} (cf. le tableau principal 7.8) et le vecteur A_8 (cf. le tableau supplémentaire 7.10). Ainsi, on peut éliminer de la base le vecteur A_{10} ou le vecteur A_8. Le vecteur A_8, en somme, n'est pas encore introduit dans la base. On n'a fait que vérifier s'il était utile d'introduire A_8. On est dans le cas de dégénérescence. Pour parer au cyclage, nous nous servirons de la règle donnée au § 4. Dans notre cas toutes les colonnes e_j du tableau principal 1 sont non marquées. C'est pourquoi la règle rendant impossible le cyclage est appliquée ici de la façon suivante.

Imaginons qu'on ait écrit dans le tableau principal 7.8, au lieu de la $(m+1)$-ième ligne, la ligne $A_r = A_8$, dont tous les éléments (dans les colonnes e_j) sont égaux à $e_{rj} = e_{8j} = 0$. Comparons les rapports des éléments de la colonne e_1 figurant dans les lignes A_{10} et A_8 aux éléments respectifs de la colonne A_8. Les deux rapports sont nuls, et il n'y a pas encore de raisons de préférer l'un des vecteurs à l'autre. Comparons maintenant les rapports des éléments respectifs des colonnes e_2 et A_8. On a

$$\frac{e_{10,2}}{x_{10,8}} = \frac{1}{4}, \qquad \frac{e_{8,2}}{x_{8,8}} = \frac{0}{-1} = 0.$$

Le plus petit rapport correspond au vecteur A_8. Par conséquent, le vecteur A_8 est à retrancher de la base (plus exactement, le vecteur A_8 ne doit pas être inclus dans la base).

Insistons encore une fois sur le fait que le cyclage, dans les problèmes de programmation linéaire, est un phénomène excessivement rare. C'est pourquoi, dans les calculs pratiques, on peut dans le cas dégénéré exclure de la base n'importe quel vecteur. Ceci étant, il est bon de retenir le fait qu'en éliminant le vecteur prévu pour être ajouté à la base (dans notre cas, A_8), on abrège les calculs relatifs à la composition du tableau principal successif.

Passons à la deuxième itération.

Composons le tableau principal 2 (tableau 7.11).

Nous avons vu que $B_2 \equiv B_1$, c'est-à-dire que les bases de la première et de la deuxième étape coïncident. C'est pourquoi les valeurs des e_{ij} dans le tableau principal 7.11 coïncident avec les grandeurs correspondantes du tableau principal 7.8:

$$e'_{ij} = e_{ij}.$$

Tableau 7.11

Tableau principal 2

B	$c_{k_j j}$	$\widetilde{X}^{(2)}$	e_1	e_2	e_3	e_4	A_6	α (β)	θ
A_9		12	1				—3	—	—
A_{10}				1			—2	—	—
A_{11}		11			1		1		11
A_{12}		5				1		—	—
	21						—1	—	—

Les composantes du programme $X^{(2)}$ se calculent à l'aide des composantes de $X^{(1)}$ suivant les formules (3.19).

Ainsi, par exemple

$$x_9' = x_9 - \theta_0 \cdot x_{9r} = x_9 - \theta_0 \cdot x_{98} = 10 - 2\,(-1) = 12,$$

$$x_{10}' = x_{10} - \theta_0 \cdot x_{10,8} = 8 - 2 \cdot 4 = 0,$$

$$x_8' = x_8 - \theta_0 \cdot x_{88} = 0 - 2\,(-1) = 2.$$

Sur la colonne $\widetilde{X}^{(2)}$ on ne porte que les composantes de base du programme.

Les quantités $c_{k_j j}$ sont comme auparavant égales à zéro (puisque la base se compose des mêmes vecteurs qu'à la première étape des calculs).

Le dernier élément de la colonne $c_{k_j j}$ est égal à la valeur de la fonction F sur le programme $X^{(2)}$. La valeur de $F\,(X^{(2)})$ se calcule selon la formule de récurrence

$$F\,(X^{(2)}) = F\,(X^{(1)}) + \theta_0 \Delta_\nu + \sum_{i=1}^{\nu} \delta_i \theta_{\gamma_i j_i}.$$

Dans notre cas, $F\,(X^{(1)}) = 0$, $\theta_0 = 2$. Les valeurs des autres paramètres sont prises dans le tableau supplémentaire 7.10:

$$F\,(X^{(2)}) = 0 + 2\Delta_2 + (\delta_1\theta_{18} + \delta_2\theta_{28}) = 0 + 2 \cdot 8 + (1 \cdot 1 + 2 \cdot 2) = 21.$$

Dans le tableau principal 2, tout $\lambda_i' = 0$ (puisque tout $c_{k_j j}' = 0$).

Complétons le tableau auxiliaire par des lignes nouvelles (cf. tableau 7.12). Aux vecteurs $A_1, \ldots, A_7$, qui ne sont pas entrés dans la base, correspondent les composantes nulles du programme, et seul le vecteur A_8 correspond à la composante du programme $x_8' = 2$. Il ressort du tableau II que $x_8' = d_{28}$. C'est pourquoi, dans le tableau auxiliaire 2 (tableau 7.12), on a pour la deuxième itération

$$k_1 = k_2 = \ldots = k_7 = 0, \quad k_8 = 2.$$

La deuxième ligne du tableau n'est pas remplie: tout $z_j' = 0$, puisque $\lambda_i' = 0$. Donc, $c_{k_j j} = c_{0j} = c_j$ pour $j = 1, 2, \ldots, 7$. On tire du tableau 7.7

Tableau 7.12

Tableau auxiliaire (2-ième itération)

	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}	A_{11}	A_{12}
k_j								2	—	—	—	—
z_j									—	—	—	—
$c_{k_j j}$	8	9	8	6	8	9	8	8	—	—	—	—
$\bar{\Delta} k_{jj}$	8	9	8	6	8	$\boxed{9}$	8	8	—	—	—	—
$c_{k_{j-1}, j}$	—	—	—	—	—	—	—	—	—	—	—	—
$\bar{\bar{\Delta}}_{k_j j}$	—	—	—	—	—	—	—	—	—	—	—	—

que $c_{k_8 8} = c_{28} = 8$, $\bar{\Delta}_{kjj} = c_{kjj} - z_j = c_{kjj}$. Ici $\bar{\Delta}_{kjj} > 0$ pour tout A_j n'entrant pas dans la base. Aussi n'est-il pas nécessaire de calculer les éléments des deux dernières lignes du tableau. En outre, ceci traduit le fait que $\tilde{X}^{(2)}$ n'est pas solution du problème et qu'il faut poursuivre les calculs. $\bar{\Delta}_{kjj}$ atteint le maximum avec deux valeurs de j ($j = 2$ et $j = 6$). On peut introduire dans la base soit A_2, soit A_6. Posons, pour fixer les idées, qu'on a opté pour A_6. Dans le tableau, l'élément $\bar{\Delta}_{06}$ est encadré. Le vecteur A_6 qui lui correspond peut être introduit dans la base suivante.

On introduit dans le tableau principal 7.11 la colonne A_6, aux composantes $x'_{i6} = \sum_{s=1}^{m} e'_{is} a_{s6}$. Dans notre cas, $\| e'_{ij} \|$ est une matrice unité et

$$x'_{i6} = a_{i6}.$$

Dans la dernière ligne du tableau principal 7.11, on indique dans la colonne $\tilde{X}^{(2)}$ la valeur $x_r = x_6 = 0$ et l'on place $x_{rr} = x'_{66} = -1$ dans la colonne A_6. On détermine le vecteur à éliminer de la base. Dans chacune des colonnes α (β) et θ du tableau principal 2 on ne remplit qu'une case: $\alpha_{11} = 0$ et $\theta_{11} = 11$. Par conséquent, $\theta' = \theta_{11} = 11$.

Le tableau supplémentaire 2 (tableau 7.13) se compose d'une seule colonne A_6, non subdivisée en sous-colonnes.

Ici, $\Delta_t = \Delta_1 > 0$. C'est pourquoi $\theta'' = \infty$ et $\theta_0 = \min(\theta', \theta'') = \theta' = 11$. La valeur θ_0 n'est atteinte que sur un seul vecteur. Donc, nous avons affaire au cas non dégénéré, et pour déterminer le vecteur à éliminer de la base aucun besoin n'est d'appliquer une règle supplémentaire. Il faut éliminer de la base le vecteur A_{11} tel que $\theta = \theta_0$.

Sur ce prend fin la deuxième itération. Pour passer de la deuxième à la troisième étape on fait appel aux mêmes règles qui régissent le passage des tableaux de la première étape à ceux de la deuxième étape. On donne ci-dessous

Tableau 7.13

Tableau supplémentaire 2

	A_6
$\theta_{\gamma j}$	2
γ	1
N	1
δ_t	1
Δ_t	8

les tableaux principaux (cf. tableau 7.14), supplémentaires (cf. tableau 7.15) et auxiliaire (cf. tableau 7.16) correspondant aux étapes suivantes de la résolution du problème.

Voici certaines particularités des calculs que l'on n'avait pas rencontrées dans les deux premières itérations.

Toutes les itérations, sauf l'itération 7, aboutissaient au cas 1°, c'est-à-dire que $\Delta = \max(\overline{\Delta}_{kjj}, \overline{\overline{\Delta}}_{kjj})$ était atteint sur un des nombres $\overline{\Delta}_{kjj}$. Pour la 7-ième itération, $\Delta = \overline{\overline{\Delta}}_{2,2}$. Dans ce cas, la colonne α (β) du tableau principal est composée selon les règles suivantes.

Si $x_{jr} = x_{j2} > 0$, on porte β_j sur la case correspondante de la colonne α (β). Si $x_{jr} < 0$, la valeur qui lui correspond dans la colonne α (β) est α_j.

Les cases de la colonne α (β) qu'il faudrait remplir par $+\infty$ ou $-\infty$ et les cases pour lesquelles $x_{jr} = 0$ sont, comme dans le cas 1°, marquées par un trait.

Dans la colonne α (β) du tableau principal répondant à l'itération 7, on ne remplit que les première, quatrième et cinquième cases avec respectivement les valeurs $\alpha_9 = 0$, $\beta_7 = 7$ et $\alpha_r = \alpha_2 = 0$.

Dans le cas 2°, les éléments de la colonne θ se calculent comme rapports de la différence entre les éléments des colonnes α (β) et $\widetilde{X}$ et les éléments respectifs de la colonne A_r.

Dans notre cas, $\theta_9 = \frac{77}{3}$, $\theta_7 = 0$, $\theta_2 = 2$,

$$\theta' = \min\left(\frac{77}{3},\ 0,\ 2\right) = 0.$$

Pour la 7-ième itération, nul besoin n'est de former le tableau supplémentaire, puisqu'il est clair que dans ce cas il n'y a pas de valeurs de $\theta_{\gamma j} \leqslant \theta' = 0$.

A la 7-ième itération, $\theta_0 = \theta' = 0$. C'est pourquoi le passage au programme suivant ne conduit pas à un accroissement de la fonction F.

Dans notre exemple, à toutes les itérations $\theta_0 = \theta'$, ce qui veut dire que dans les tableaux supplémentaires tout Δ_t est positif. Ceci signifie que la valeur limite de la composante du vecteur à ajouter à la base est définie par les contraintes du problème ($\alpha_j \leqslant x_j \leqslant \beta_j$).

Tableau 7.14

Tableaux principaux des itérations 3 à 10

N° de l'itér.	B	$c_{k_j j}$	$\widetilde{X}^{(3)}$	e_1	e_2	e_3	e_4	A_1	α (β)	θ
3	A_9		45	1		3		—5	—	—
	A_{10}		22		1	2		—1	—	—
	A_6	8	11			1		—2	—	—
	A_{12}		5				1	5		1
		111				8		—1	4	4
N° de l'itér.	B	$c_{k_j j}$	$\widetilde{X}^{(4)}$	e_1	e_2	e_3	e_4	A_8	α (β)	θ
4	A_9		50	1		3	1	—7	—	—
	A_{10}		23		1	2	1/5		—	—
	A_6	8	13			1	2/5	—2	—	—
	A_1	5	1				1/5		—	—
		135	2			8	21/5	—1	5	3
N° de l'itér.	B	$c_{k_j j}$	$\widetilde{X}^{(5)}$	e_1	e_2	e_3	e_4	A_2	α (β)	θ
5	A_9		71	1		3	1	—3	—	—
	A_{10}		23		1	2	1/5	11/5		115/11
	A_6	8	19			1	2/5	—3/5	—	—
	A_1	5	1				1/5	—4/5	4	15/4
		200				8	21/5	—1	5	5
N° de l'itér.	B	$c_{k_j j}$	$\widetilde{X}^{(6)}$	e_1	e_2	e_3	e_4	A_7	α (β)	θ
6	A_9		329/4	1		3	1/4	3/4		329/3
	A_{10}		59/4		1	2	3/4	—15/4	—	—
	A_6	8	85/4			1	1/4	—1/4	—	—
	A_2	5	15/4				—1/4	1/4		15
		1019/4				8	3/4	—1	7	7

Suite

N° de l'itér.	B	c_{kj}	$\widetilde{X}^{(7)}$	e_1	e_2	e_3	e_4	A_2	α (β)	θ
7	A_9		77	1		3	1	—3		77/3
	A_{10}		41		1	2	—3	15	—	—
	A_6	8	23			1		1	—	—
	A_7	5	7				—1	4	7	
		304	2			8	—5	—1		2
N° de l'itér.	B	c_{kj}	$\widetilde{X}^{(8)}$	e_1	e_2	e_3	e_4	A_4	α (β)	θ
8	A_9		77	1		3	1/4	—1/4	—	—
	A_{10}		41		1	2	3/4	17/4		164/17
	A_6	8	23			1	1/4	3/4		92/3
	A_2	7	2				—1/4	—3/4	5	4
		304				8	1/4	—1	—	—
N° de l'itér.	B	c_{kj}	$\widetilde{X}^{(9)}$	e_1	e_2	e_3	e_4	A_5	α (β)	θ
9	A_9		78	1		3	1/3	22/3		117/11
	A_{10}		24		1	2	—2/3	16/3		9/2
	A_6	8	20			1		1		20
	A_4	6	4				1/3	—2/3	—	—
		318				8	2	—1	2	2
N° de l'itér.	B	c_{kj}	$\widetilde{X}^{(10)}$	e_1	e_2	e_3	e_4			
10	A_9'		190/3	1		3	1/3			
	A_{10}		40/3		1	2	—2/3			
	A_6	8	18			1				
	A_4	6	16/3				1/3			
		323				8	2			

Tableau 7.15

Tableaux supplémentaires des itérations 3 à 10

		A_1	
3	$\theta_{\gamma j}$	1	$\theta' = \min\ (1,4) = 1;$
	γ	1	$\theta'' = \infty,\ \theta_0 = \theta' = 1.$
	N	1	
	δ_t	3	
	Δ_t	21	

		A_8		
4	$\theta_{\gamma j}$	1	2	$\theta' = 3;\ \theta'' = \infty;$
	γ	3	4	$\theta_0 = \theta' = 3.$
	N	1	2	
	δ_t	3	1	
	Δ_t	21	20	

		A_1		A_2			
5	$\theta_{\gamma j}$	5/4	5/2	1	2	3	$\theta' = \min \left\{ \frac{115}{11}, \frac{15}{4}, 5 \right\} = \frac{15}{4},$
	γ	2	3	1	2	3	
	N	2	4	1	3	5	
	δ_t	4/5	8/5	2	1	1	$\theta'' = \infty,\ \theta_0 = \theta' = \frac{15}{4}.$
	Δ_t	15	62/5	79/5	14	57/5	

Suite

		A_7		A_2		
6	$\theta_{\gamma j}$	2	4	3	7	
	γ	1	2	3	2	
	N	1	3	2	4	$\theta' = \min\left\{\frac{329}{3}, 15, 7\right\} = 7;$
	δ_t	1	2	1/4	1/4	$\theta'' = \infty;\ \theta_0 = \theta'' = 7.$
	Δ_t	31/4	11/2	15/2	21/4	

7	$\theta_0 = \theta' = 0.$ Le tableau supplémentaire n'est pas nécessaire.

		A_2			
8	$\theta_{\gamma j}$	0	4/3	8/3	$\theta' = \min\left\{\frac{164}{17}, \frac{93}{3}, 4\right\} = 4;$
	γ	2	3	4	
	N	1	2	3	
	δ_t	3/4	3/4	3/2	$\theta'' = \infty,\ \theta_0 = \theta' = 4.$
	Δ_t	9/2	15/4	9/4	

		A_5	
9	$\theta_{\gamma j}$	1	$\theta' = \min\left\{\frac{117}{11}, \frac{9}{2}, 20, 2\right\} = 2;$
	γ	1	
	N	1	
	δ_t	3	$\theta'' = \infty,\ \theta_0 = \theta' = 2.$
	Δ_t	1	

Le programme optimal du problème non linéaire (6.5), (6.2), (6.6), équivalent au problème de programmation linéaire (6.1)-(6.4) est déterminé d'après la colonne $\widetilde{X}^{(10)}$ du tableau principal 10 et d'après les numéros k_j des valeurs critiques tirées du tableau supplémentaire associé à la dernière itération.

On a

$$X = \left(4, 5, 0, \frac{16}{3}, 2, 18, 7, 5\right). \tag{6.7}$$

Nous avons omis ici les valeurs des variables additionnelles.

Tableau 7.16

Tableaux auxiliaires des itérations 3 à 10

N° de l'itér.		A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}	A_{11}	A_{12}
3	k_j						—		2	—	—		—
	z_j	—16	8	8		8	—		—16	—	—	8	—
	c_{kjj}	8	9	8	6	8	—	8	8	—	—		—
	$\overline{\Delta}_{kjj}$	$\boxed{24}$	1		6		—	8	24	—	—	—8	—
	$c_{kj-1,j}$	—	—	—	—	—	—	—	—	—	—	—	—
	$\overline{\overline{\Delta}}_{kjj}$	—	—	—	—	—	—	—	—	—	—	—	—
4	k_j	—					—		2	—	—		
	z_j	—	—44/5	8	63/5	—2/5	—	—21/5	—16	—	—	8	21/5
	c_{kjj}	—	9	8	6	8	—	8	8	—	—		
	$\overline{\Delta}_{kjj}$	—	89/5		—33/5	42/5	—	61/5	$\boxed{24}$	—	—	—8	—21/5
	$c_{kj-1,j}$	—	—	—	—	—	—	—	—	—	—	—	—
	$\overline{\overline{\Delta}}_{kjj}$	—	—	—	—	—	—	—	—	—	—	—	—
5	k_j	—					—		5	—			
	z_j	—	—44/5	8	63/5	—2/5	—	—21/5	—16	—	—	8	21/5
	c_{kjj}	—	9	8	6	8	—	8	—	—	—		
	$\overline{\Delta}_{kjj}$	—	$\boxed{89/5}$		—23/5	42/5	—	61/5	—	—	—	—8	—21/5
	$c_{kj-1,j}$	—	—	—	—	—	—	—	4	—	—	—	—
	$\overline{\overline{\Delta}}_{kjj}$	—	—	—	—	—	—	—	—20	—	—	—	—

Suite

N° de l'itér.		A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}	A_{11}	A_{12}
6	k_j	4	—				—		5	—	—		
	z_j	—49/4	—	8	9/4	13/2	—	—3/4	—16	—	—	8	3/4
	c_{hjj}	—	—	8	6	8	—	8	—	—	—		
	$\overline{\Delta}_{hjj}$	—	—		15/4	3/2	—	$\boxed{35/4}$	—	—	—	—8	—3/4
	$c_{hj-1,\ j}$	2	—	—	—	—	—	—	4	—	—	—	—
	$\overline{\overline{\Delta}}k_{jj}$	—57/4	—	—	—	—	—	—	—20	—	—	—	—
7	k_j	4	2				—	—	5	—	—		
	z_j	—41	28	8	—15	18	—	—	—16	—	—	8	—5
	c_{hjj}	—	6	8	6	8	—	—	—	—	—		
	$\overline{\Delta}_{hjj}$	—	—22		21	—10	—	—	—	—	—	—8	5
	$c_{hj-1,j}$	2	$\boxed{7}$	—	—	—	—	—	4	—	—	—	—
	$\overline{\overline{\Delta}}k_{jj}$	—43	21	—	—	—	—	—	—20	—	—	—	—
8	k_j	4	—				—	3	5	—	—		
	z_j	—59/4	—	8	3/4	15/2	—	—1/4	—16	—	—	8	1/4
	c_{hjj}	—	—	8	6	8	—	—	—	—	—		
	$\overline{\Delta}_{hjj}$	—	—		$\boxed{21/4}$	1/2	—	—	—	—	—	—8	—1/4

	$c_{kj-1,\ j}$	2	—	—	—	—	—	5	4	—	—	—	—
	$\overline{\overline{\Delta}}_{kjj}$	—67/4	—	—	—	—	—	—21/4	—20	—	—	—	—
	k_j	4	5		—		—	3	5	—	—		
	z_j	—6		8	—	4	—	—2	—16	—	—	8	2
	c_{kjj}	—	—	8	—	8	—	—	—	—	—		
9	$\overline{\Delta}_{kjj}$	—	—		—	$\boxed{4}$	—	—	—	—	—	—8	—2
	$c_{kj-1,\ j}$	2	3	—	—	—	—	5	4	—	—	—	—
	$\overline{\overline{\Delta}}_{kjj}$	—8	—3	—	—	—	—	—7	—20	—	—	—	—
	k_j	4	5		—	2	—	3	5	—	—		
	z_j	—6		8	—	4	—	—2	—16	—	—	8	2
	c_{kjj}	—	—	8	—	—	—	—	—	—	—		
10	$\overline{\Delta}_{kjj}$	—	—		—	—	—	—	—	—	—	—8	—2
	$c_{kj-1,j}$	2	3	—	—	5	—	5	4	—	—	—	—
	$\overline{\overline{\Delta}}_{kjj}$	—8	—3	—	—	—1	—	—7	—20	—	—	—	—

Les composantes x_i du programme optimal du problème de programmation linéaire (6.1)-(6.4) coïncident avec les composantes du vecteur X qui définit la solution du problème non linéaire équivalent [cf. (6.7)].

Les valeurs optimales des variables $x_j^{(i)}$ se calculent à l'aide des composantes x_j de la solution d'après la formule

$$x_j^{(i)} = \begin{cases} x_j - d_{ij} & \text{pour} \quad x_j > d_{ij}, \\ 0 & \text{pour} \quad x_j \leqslant d_{ij}. \end{cases}$$

Dans le cas considéré on a:

Tableau 7.17

Composantes $x_j^{(i)}$ de la solution

$i \backslash j$	1	2	3	4	5	6	7	8
1	3	4	0	—	1	16	5	4
2	2	3	0	—	—	—	3	3
3	1	2	—	—	—	—	0	2
4	—	1	—	—	—	—	—	1

La valeur maximale de la forme linéaire (6.1) coïncide avec la valeur optimale de la fonction F; d'après le dernier tableau principal, elle est égale à 323.

Constatons que le problème (6.1)-(6.4), étant ramené à la forme canonique, est un problème de programmation linéaire à 52 variables non négatives et 27 conditions-égalités. L'algorithme décrit au § 5 a permis d'obtenir la solution de ce problème en 10 itérations, bien que le programme initial choisi fût manifestement défavorable (pour la base initiale, $F = 0$). Le volume de calculs à fournir à chaque itération est à peu près le même que dans les problèmes de programmation linéaire à 4 conditions-égalités et 12 variables non négatives.

Pour résoudre le problème (6.1)-(6.4) par l'une quelconque des méthodes générales de programmation linéaire le nombre d'itérations à réaliser est nettement plus grand, et le volume de calculs à effectuer à chaque itération augmente considérablement.

§ 7. Généralités sur la construction des algorithmes de programmation linéaire par morceaux

7.1. Les §§ 3 à 6 donnent une description détaillée de la méthode et de l'algorithme de résolution du problème I. Considérons maintenant les principes qui sont à la base des algorithmes de résolution d'autres problèmes de programmation linéaire par morceaux énoncés au § 1. Commençons par le problème V, qui diffère du problème I par la présence dans sa fonction économique d'une somme de fonctions linéaires et de plusieurs superpositions de fonctions linéaires par morceaux d'une variable. Le problème V consiste à minimiser

la fonction linéaire par morceaux convexe vers le bas

$$F(X)=\sum_{j=1}^{n} f_j(x_j)+\sum_{\mu=n+1}^{n+s} f_\mu(L_\mu(X)) \tag{7.1}$$

sous les conditions

$$\sum_{j=1}^{n} a_{ij}x_j=b_i, \quad i=1,\ 2,\ \ldots,\ m, \tag{7.2}$$

$$\alpha_j \leqslant x_j \leqslant \beta_j, \quad j=1,\ 2,\ \ldots,\ n. \tag{7.3}$$

Les fonctions convexes vers le bas $f_j(x_j)$, $j=1, 2, \ldots, n+s$, sont définies ici par la formule (3.6), $-\alpha_j=\beta_j=\infty$ pour $j \geqslant n+1$; les fonctions linéaires $L_\mu(X)$ s'écrivent

$$L_\mu(X)=\sum_{j=1}^{n} a_{\mu j}x_j-b_\mu, \quad \mu=n+1,\ \ldots,\ n+s.$$

Il est évident que le problème (7.1)-(7.3) se ramène facilement au problème du type (3.1)-(3.3); il n'y a qu'à introduire les variables additionnelles

$$x_{n+i}=\sum_{j=1}^{n} a_{n+i,j}x_j-b_{n+i}, \quad i=1,\ 2,\ \ldots,\ s.$$

C'est pourquoi l'algorithme de résolution du problème (7.1)-(7.3) repose sur la méthode décrite au § 3. Pour obtenir un procédé économique d'analyse du problème (7.1)-(7.3) il faut seulement tenir compte (comme cela a été fait aux §§ 1, 2 du chapitre 11 de [87]) que les vecteurs des conditions associés aux variables additionnelles x_{n+i} sont unitaires. En se fondant sur la notion de programme de base du problème (3.1)-(3.3), on aboutit à la définition suivante.

Un programme $X=(x_1, x_2, \ldots, x_n)$ du problème (7.1)-(7.3) est dit de base s'il lui correspond une matrice carrée non dégénérée A_X de la forme

$$A_X=\begin{Vmatrix} a_{1j_1} & a_{1j_2} & \ldots & a_{1j_{m+t}} \\ a_{2j_1} & a_{2j_2} & \ldots & a_{2j_{m+t}} \\ \cdots & \cdots & \cdots & \cdots \\ a_{mj_1} & a_{mj_2} & \ldots & a_{mj_{m+t}} \\ a_{n+\lambda_1,j_1} & a_{n+\lambda_1,j_2} & \ldots & a_{n+\lambda_1,j_{m+t}} \\ \cdots & \cdots & \cdots & \cdots \\ a_{n+\lambda_t,j_1} & a_{n+\lambda_t,j_2} & \ldots & a_{n+\lambda_t,j_{m+t}} \end{Vmatrix}, \tag{7.4}$$

telle que:

a) avec $j \neq j_\alpha$ ($\alpha=1, 2, \ldots, m+t$), la composante x_j du programme X coïncide avec l'une de ses valeurs critiques;

b) chaque variable additionnelle $x_{n+\lambda_\alpha} = L_{n+\lambda_\alpha}(X)$, $\alpha = 1, 2, \ldots, t$ $(0 \leqslant t \leqslant \min\{n-m, s\})$ soit égale à l'une de ses valeurs critiques.

La matrice (7.4) s'appelle base du programme X.

Le programme de base X est dit non dégénéré si aucune de ses composantes x_{j_α} $(\alpha = 1, 2, \ldots, m+t)$ ni de ses variables additionnelles $x_{n+i} = L_{n+i}(X)$ $(i \neq \lambda_\alpha,\ \alpha = 1, 2, \ldots, t)$ ne coïncident avec l'une de ses valeurs critiques.

Un programme de base qui ne jouit pas de cette propriété est considéré comme dégénéré.

Dans ce qui suit, le problème (7.1)-(7.3) est supposé non dégénéré, c'est-à-dire dépourvu de programmes de base dégénérés.

La procédure de résolution du problème (7.1)-(7.3) se compose d'un nombre fini d'itérations. Une itération élémentaire compte, comme d'habitude, deux étapes. La première consiste à tester l'optimalité du programme de base disponible, et la deuxième à améliorer le programme dont l'optimalité n'a pas été établie à la première étape. Voici la description de la suite d'opérations constituant une itération de l'algorithme. Pour simplifier l'écriture des indices, posons que pour le programme de base initial X de l'itération considérée $j_\alpha = \alpha$, $\alpha = 1, 2, \ldots, m+t$, $\lambda_\alpha = \alpha$, $\alpha = 1, 2, \ldots, t$, c'est-à-dire que la base (7.4) du programme X est de la forme

$$A_X = \begin{Vmatrix} a_{11} & a_{12} & \ldots & a_{1.\,m+t} \\ a_{21} & a_{22} & \ldots & a_{2.\,m+t} \\ \cdots & \cdots & \cdots & \cdots \\ a_{m1} & a_{m2} & \ldots & a_{m.\,m+t} \\ a_{n+1.1} & a_{n+1.2} & \ldots & a_{n+1.\,m+t} \\ \cdots & \cdots & \cdots & \cdots \\ a_{n+t.1} & a_{n+t.2} & \ldots & a_{n+t.\,m+t} \end{Vmatrix}.$$

7.2. Avant de commencer une itération, on dispose d'un programme de base X tel que, par suite des hypothèses faites plus haut, il justifie les relations

$$\left.\begin{aligned} d_{k_j j} < x_j < d_{k_j+1,\,j}, \quad & j = 1, 2, \ldots, m+t, \\ x_j = d_{k_j j}, \quad & m+t < j \leqslant n, \\ d_{k_j j} < L_j(X) < d_{k_j+1,\,j}, \quad & j > n+t, \\ L_j(X) = d_{k_j,\,j}, \quad & n+1 \leqslant j \leqslant n+t, \end{aligned}\right\}. \tag{7.5}$$

et la matrice A_X^{-1} inverse de la base A_X.

Posons

$$c_j^{(X)} = c_{k_j j} + \sum_{\lambda=n+t+1}^{n+s} a_{\lambda j} c_{k_\lambda \lambda}, \quad j = 1, 2, \ldots, m+t,$$

$$C_X = (c_1^{(X)}, c_2^{(X)}, \ldots, c_{m+t}^{(X)}).$$

Soit

$$\Lambda = (\lambda_1, \lambda_2, \ldots, \lambda_m, -\lambda_{n+1}, -\lambda_{n+2}, \ldots, -\lambda_{n+t})$$

le vecteur vérifiant le système d'équations

$$\Lambda A_X = C_X,$$

c'est-à-dire

$$\Lambda = C_X A_X^{-1}.$$

Le vecteur Λ se calcule à la première étape de l'itération et sert à vérifier si le programme de base X est optimal.

Les conditions (2.53)-(2.56) du critère d'optimalité d'un programme du problème (7.1)-(7.3) peuvent s'écrire pour le cas considéré

$$c_{k_j-1,\,j} \leqslant \sum_{i=1}^{m} a_{ij}\lambda_i - \sum_{i=n+1}^{n+t} a_{ij}\lambda_i - \sum_{i=n+t+1}^{n+s} a_{ij}c_{k_i i} \leqslant c_{k_j j},$$

$$m+t < j \leqslant n, \tag{7.6}$$

$$c_{k_j-1,\,j} \leqslant \lambda_j \leqslant c_{k_j j}, \quad n+1 \leqslant j \leqslant n+t. \tag{7.7}$$

De même qu'au § 3, ici

$$c_{-1j} = -\infty, \quad c_{l_j+1,\,j} = \infty \quad (j = 1, 2, \ldots, n).$$

Si les relations (7.6), (7.7) sont remplies, le programme X est la solution recherchée du problème. Mais si l'une au moins de ces relations n'est pas respectée, il faut passer à la deuxième étape de l'itération, c'est-à-dire améliorer le programme disponible. Supposons que les conditions d'optimalité (7.6)-(7.7) soient compromises. Cela traduit le fait que parmi les nombres

$$\bar{\Delta}_{k_j j} = \begin{cases} c_{k_j j} - \sum\limits_{i=1}^{m} a_{ij}\lambda_i + \sum\limits_{i=n+1}^{n+t} a_{ij}\lambda_i + \sum\limits_{i=n+t+1}^{n+s} a_{ij}c_{k_i i} \\ \qquad \text{si } m+t < j \leqslant n, \\ c_{k_j j} - \lambda_j \text{ si } n+1 \leqslant j \leqslant n+t, \end{cases}$$

et

$$\bar{\bar{\Delta}}_{k_j j} = -\bar{\Delta}_{k_j j} + \sigma_{k_j j} \quad (m+t < j \leqslant n+t)$$

il en est qui sont négatifs. Choisissons l'un de ces nombres (le plus petit, par exemple). Dans ces conditions on peut se trouver devant l'une des quatre éventualités suivantes:

a) le choix concerne la quantité $\bar{\Delta}_{k_r r}$, $m+t < r \leqslant n$, associée au deuxième membre de la r-ième condition du système (7.6);

b) le choix concerne la quantité $\bar{\bar{\Delta}}_{k_r r}$, $m+t < r \leqslant n$, associée au premier membre de la r-ième condition du système (7.6);

c) le nombre choisi est $\Delta_{k_r r}$, $n+1 \leqslant r \leqslant n+t$, associé au deuxième membre de la r-ième condition du système (7.7);

d) le nombre choisi est $\bar{\bar{\Delta}}_{k_r r}$, $n+1 \leqslant r \leqslant n+t$, associé au premier membre de la r-ième condition du système (7.7).

7.3. Explorons séparément chacune de ces éventualités.

a) Dans le premier cas, où

$$\bar{\Delta}_{k_r r} < 0, \quad m+t < r \leqslant n,$$

on introduit le vecteur $X(\theta)$ dont les composantes $x_j(\theta)$ se calculent d'après la formule

$$x_j(\theta) = \begin{cases} x_j - \theta x_{jr}, & 1 \leqslant j \leqslant m+t, \\ x_j + \theta, & j = r, \\ x_j, & j \neq r, \quad m+t < j \leqslant n. \end{cases} \tag{7.8}$$

Le vecteur $X_r = (x_{1r}, x_{2r}, \ldots, x_{m+t,r})^T$ est ici composé des coefficients de décomposition du vecteur $A_r = (a_{1r}, \ldots, a_{mr}, a_{m+1,r}, \ldots, a_{m+t,r})^T$ par rapport aux vecteurs-colonnes de la matrice A_X, soit

$$X_r = A_X^{-1} A_r.$$

On choisit le paramètre $\theta \geqslant 0$ de la formule (7.8), comme au § 3, en partant de deux restrictions. La première d'entre elles impose que $X(\theta)$ soit un programme du problème, c'est-à-dire qu'il satisfasse aux conditions (7.3). Pour respecter cette condition, le paramètre θ doit être compris entre 0 et θ', où θ' est le plus petit des trois nombres

$$\min_{x_{jr}>0} \left(\frac{x_j - \alpha_j}{x_{jr}}\right), \quad \min_{x_{jr}<0} \left(\frac{x_j - \beta_j}{x_{jr}}\right), \quad \beta_r - x_r. \tag{7.9}$$

Dans les relations (7.9), l'indice j parcourt les valeurs $1, 2, \ldots, m+t$. La deuxième restriction limite le choix de θ au secteur de décroissance monotone de la fonction $F(X(\theta))$ lorsque θ croît. L'extrémité gauche de ce secteur (segment ou demi-droite) est 0.

Pour déterminer la borne droite θ'' du secteur de décroissance monotone de la fonction $F(X(\theta))$, nous procédons de la même façon qu'au § 3. Calculons les valeurs

$$\theta_{\gamma j} = \frac{x_j - d_{\gamma j}}{x_{jr}}$$

pour $j = 1, 2, \ldots, m+t, r, n+t+1, \ldots, n+s$ $(x_{jr} \neq 0)$ et pour $\gamma \neq 0, l_j + 1$, tels que

$$0 \leqslant \theta_{\gamma j} < \theta'.$$

De plus, pour $j > n$ on entend évidemment par x_{jr}

$$x_{jr} = \frac{x_j - x_j(\theta)}{\theta} = \frac{L_j(X) - L_j(X(\theta))}{\theta} =$$

$$= \sum_{\mu=1}^{m+t} a_{j\mu} x_{\mu r} - a_{jr}; \quad x_{rr} = -1. \tag{7.10}$$

Rangeons les valeurs obtenues de $\theta_{\nu j}$ par ordre de croissance.

On a

$$\theta_{\nu_1 j_1} < \theta_{\nu_2 j_2} < \ldots < \theta_{\nu_p j_p}. \tag{7.11}$$

Soit

$$\Delta_q = \bar{\Delta}_{k_r r} + \sum_{i=1}^{q} \sigma_{\nu_i j_i} | x_{j_i r} | \, (\Delta_0 = \bar{\Delta}_{k_r r}).$$

La borne θ'' est posée égale à $\theta_{\nu_\nu j_\nu}$ si

$$\Delta_{\nu-1} < 0, \quad \Delta_\nu \geqslant 0 \quad (\nu = 1, 2, \ldots, p).$$

Mais si $\Delta_p < 0$, alors $\theta'' = \infty$.

Après le calcul de θ' et de θ'', posons

$$\theta_0 = \min \{\theta', \theta''\}.$$

Dans le cas $\theta_0 = \infty$, la solution du problème (7.1)-(7.3) est irréalisable la fonction (7.1) étant infinie sur l'ensemble des programmes du problème. Supposons que $\theta_0 < \infty$. Le vecteur $X' = X(\theta_0)$ est alors le programme de base du problème (7.1)-(7.3) associé à une valeur plus petite de la fonction (7.1) par rapport aux autres programmes du type $X(\theta)$.

b) Dans le deuxième cas, où l'on choisit le nombre

$$\bar{\bar{\Delta}}_{k_r r} < 0, \quad m + t < r \leqslant n,$$

le programme X est amélioré par le passage au programme $X(\theta)$ aux composantes $x_j(\theta)$ calculées selon la formule

$$x_j(\theta) = \begin{cases} x_j + \theta x_{jr}, & 1 \leqslant j \leqslant m + t, \\ x_j - \theta, & j = r, \\ x_j, & j \neq r, \quad m + t < j \leqslant n. \end{cases} \tag{7.12}$$

Le calcul de la valeur θ_0 du paramètre θ de la formule (7.12) est guidé sur les mêmes considérations que dans le cas a).

7.4. Considérons les deux cas restants, où l'indice r est compris entre $n + 1$ et $n + t$, ce qui correspond à la r-ième condition du système (7.7).

c) Supposons que pour $n + 1 \leqslant r \leqslant n + t$ la quantité $\bar{\Delta}_{k_r r} < 0$. Introduisons le vecteur $X(\theta)$ aux composantes $x_j(\theta)$ définies

par la formule

$$x_j(\theta) = \begin{cases} x_j + \theta e_{jr}, & 1 \leqslant j \leqslant m + t, \\ x_j, & m + t < j \leqslant n. \end{cases} \tag{7.12'}$$

e_{jr} désigne ici, comme d'habitude, l'élément associé à la matrice inverse A_X^{-1} de la base courante A_X. La transformation (7.12′) conserve les valeurs des variables additionnelles x_j avec $j = n + 1, \ldots, n + r - 1, n + r + 1, \ldots, n + t$; $x_r(\theta) = x_r + \theta$.

En ce qui concerne les variables additionnelles x_j avec $j > n + t$, posons

$$e_{rj} = \frac{L_j(X(\theta)) - L_j(X)}{\theta} = \sum_{\lambda=1}^{m+t} a_{j\lambda} e_{\lambda r}.$$

Le vecteur $X(\theta)$ reste le programme du problème pour ceux des $\theta \geqslant 0$ pour lesquels les conditions (7.3) continuent d'être respectées. En utilisant (7.12′), on aboutit à la valeur-limite de θ', égale au plus petit des deux nombres

$$\min_{e_{jr}<0} \left(\frac{\alpha_j - x_j}{e_{jr}}\right), \quad \min_{e_{jr}>0} \left(\frac{\beta_j - x_j}{e_{jr}}\right), \quad \text{où } 1 \leqslant j \leqslant m + t.$$

Avec $\theta > 0$ qui se trouvent à gauche de θ', le vecteur $X(\theta)$ est un programme du problème (7.1)-(7.3); avec θ situés à droite de θ', certaines composantes du vecteur $X(\theta)$ deviennent négatives, et $X(\theta)$ cesse d'être un programme du problème. Pour déterminer la borne θ'' jusqu'à laquelle la fonction $F(X(\theta))$ est monotone décroissante lorsque θ croît, calculons les valeurs

$$\theta_{\gamma j} = \frac{d_{\gamma j} - x_j}{e_{jr}}$$

avec $j = 1, 2, \ldots, m + t, r, n + t + 1, \ldots, n + s$ $(e_{jr} \neq 0)$ et pour $\gamma \neq 0, l_j + 1$, tels que

$$0 \leqslant \theta_{\gamma j} < \theta'.$$

Disposons les nombres ainsi trouvés par ordre de croissance:

$$\theta_{\gamma_1 j_1} < \theta_{\gamma_2 j_2} < \ldots < \theta_{\gamma_p j_p}. \tag{7.13}$$

Soit

$$\Delta_q = \bar{\Delta}_{k_r r} + \sum_{i=1}^{q} \sigma_{\gamma_i j_i} |e_{j_i r}|.$$

Dans le secteur $\theta_{\gamma_q j_q} \leqslant \theta \leqslant \theta_{\gamma_{q+1} j_{q+1}}$ la fonction $F(X(\theta))$ est linéaire, son coefficient angulaire est Δ_q. C'est pourquoi la borne θ'' recherchée est définie de la même façon que dans les cas précédents. Avec $\Delta_p < 0$, posons $\theta'' = \infty$. Mais si $\Delta_{\nu-1} < 0$,

$\Delta_\nu \geqslant 0$ ($\nu = 1, 2, \ldots, p$), alors

$$\theta'' = \theta_{\gamma_\nu j_\nu}.$$

Les nombres θ' et θ'' permettent d'obtenir $\theta_0 = \min\{\theta', \theta''\}$.

Si $\theta_0 < \infty$, posons $X' = X(\theta_0)$. Dans le cas contraire ($\theta_0 = \infty$), on tire la conclusion que le problème n'a pas de solution.

d) Le dernier cas à explorer est celui qui apparaît lorsque $n + 1 \leqslant r \leqslant n + t$ et $\bar{\bar{\Delta}}_{k_r r} < 0$. Dans ce cas, la diminution de la fonction économique F s'obtient en passant au programme $X(\theta_0)$ dont les composantes $x_j(\theta_0)$ sont calculées d'après la formule

$$x_j(\theta_0) = \begin{cases} x_j - \theta_0 e_{jr}, & 1 \leqslant j \leqslant m + t, \\ x_j, & m + t < j \leqslant n. \end{cases} \tag{7.14}$$

Le nombre θ_0 qui figure dans la formule (7.14) est déterminé à partir des mêmes relations que dans le cas c).

7.5. L'algorithme de résolution du problème (7.1)-(7.3) est fondé sur les formules de récurrence qui servent pour transformer la matrice inverse de la base du programme de base courant.

Dans les cas a) et b), on s'efforce d'inclure dans la base la colonne de numéro r ($m + t < r \leqslant n$). Supposons que le nombre θ_0 soit atteint avec l'indice k ($k = 1, 2, \ldots, m + t, r, n + t + 1, \ldots, n + s$), c'est-à-dire

$$\theta_0 = \frac{x_k - d_{\gamma k}}{x_{kr}}$$

avec un certain $0 \leqslant \gamma \leqslant l_j + 1$.

Avec $k = r$, la base ne change évidemment pas. Si $1 \leqslant k \leqslant m + t$, la k-ième colonne quitte la base. Par conséquent, dans ce cas, la base du programme de base X' diffère de la base A_X par un seul vecteur. Pour recalculer les éléments de la matrice inverse de la base, il suffit d'appliquer les formules de récurrence usuelles de la programmation linéaire.

Enfin, avec $n + t + 1 \leqslant k \leqslant n + s$, on ajoute à la nouvelle base la ligne de numéro k. De cette façon, la matrice $A_{X'}$ se forme dans le cas considéré en bordant la matrice A_X d'une ligne et d'une colonne. La matrice inverse $A_{X'}^{-1}$ se calcule selon les formules de récurrence (2.12), (2.13) données au chapitre 11 de [87].

En présence du cas c) ou d), on s'efforce encore de retrancher de la base la ligne de numéro r ($r = n + 1, \ldots, n + t$). Le nombre θ_0 est atteint de nouveau avec l'un des indices k ($k = 1, 2, \ldots, m + t, r, n + t + 1, \ldots, n + s$). Avec $k = r$, la base ne change pas. Si $n + t + 1 \leqslant k \leqslant n + s$, la place de la r-ième ligne est occupée par la k-ième. L'ordre de la base est conservé. Les éléments de la matrice $A_{X'}^{-1}$ se calculent d'après les formules de récurrence habituelles de la programmation linéaire. Si par contre $1 \leqslant k \leqslant m + t$, on exclut de la base la k-ième colonne. Ainsi, la

matrice $A_{X'}$ se forme à partir de A_X en rayant une ligne et une colonne. Les éléments de la matrice inverse $A_{X'}$ se calculent à l'aide des formules de récurrence (2.14), (2.15) données au chapitre 11 de [87]. Nous voyons que l'algorithme de résolution du problème (7.1)-(7.3) repose sur le schéma de calcul des problèmes de programmation linéaire dont l'écriture a une forme arbitraire (§ 1, chapitre 11 de [87]), de même que l'algorithme de résolution du problème (3.1)-(3.3) est fondé sur le schéma de calcul de la méthode primale du simplexe appliquée à la forme canonique des problèmes.

7.6. Considérons maintenant le problème linéaire par morceaux II, qui consiste à minimiser la fonction économique

$$F(X)=\sum_{j=1}^{n} f_j(x_j) \tag{7.15}$$

sous les conditions

$$\sum_{j=1}^{n} g_{ij}(x_j)\leqslant 0, \quad i=1,\ 2,\ \ldots,\ m, \tag{7.16}$$

$$\alpha_j \leqslant x_j \leqslant \beta_j, \quad j=1,\ 2,\ \ldots,\ n. \tag{7.17}$$

$f_j(x_j)$ et $g_{ij}(x_j)$ sont ici des fonctions linéaires par morceaux convexes vers le bas des arguments correspondants.

Nous entendrons par valeur critique de l'argument x_j tout point du segment $[\alpha_j,\ \beta_j]$ en lequel une au moins des fonctions $f_j(x_j)$, $g_{ij}(x_j)$, $i=1,\ 2,\ \ldots,\ m$, subit une brisure.

Numérotons, selon l'usage, les valeurs critiques de l'argument x_j par ordre de croissance. On a

$$\alpha_j=d_{0j}<d_{1j}<\ldots<d_{l_j j}<d_{l_j+1,\,j}=\beta_j.$$

Les fonctions $f_j(x_j)$ et $g_{ij}(x_j)$ sont définies par les relations

$$f_j(x_j)=\begin{cases} f_j(\alpha_j)+(x_j-\alpha_j)\,c_{0j} & \text{si } \alpha_j\leqslant x_j\leqslant d_{1j} \\ f_j(d_{1j})+(x_j-d_{1j})\,c_{1j} & \text{si } d_{1j}\leqslant x_j\leqslant d_{2j}, \\ \cdots\cdots\cdots\cdots\cdots\cdots & \\ \cdots\cdots\cdots\cdots\cdots\cdots & \\ f_j(d_{l_j j})+(x_j-d_{l_j j})\,c_{l_j j} & \text{si } d_{l_j j}\leqslant x_j\leqslant \beta_j; \end{cases} \tag{7.18}$$

$$g_{ij}(x_j)=\begin{cases} g_{ij}(\alpha_j)+(x_j-\alpha_j)\,g_{0j}^{(i)} & \text{si } \alpha_j\leqslant x_j\leqslant d_{1j}, \\ g_{ij}(d_{1j})+(x_j-d_{1j})\,g_{1j}^{(i)} & \text{si } d_{1j}\leqslant x_j\leqslant d_{2j}, \\ \cdots\cdots\cdots\cdots\cdots\cdots & \\ \cdots\cdots\cdots\cdots\cdots\cdots & \\ g_{ij}(d_{l_j j})+(x_j-d_{l_j j})\,g_{l_j j}^{(i)} & \text{si } d_{l_j j}\leqslant x_j\leqslant \beta_j; \end{cases} \tag{7.19}$$

$$(i=1,\ 2,\ \ldots,\ m;\quad j=1,\ 2,\ \ldots,\ n).$$

Les fonctions $f_j(x_j)$ et $g_{ij}(x_j)$ étant convexes vers le bas, les paramètres $c_{\gamma j}$ et $g_{\gamma j}^{(i)}$ ne décroissent pas avec l'augmentation de l'indice γ, soit

$$c_{\gamma j} \leqslant c_{\gamma+1, j}, \quad g_{\gamma j}^{(i)} \leqslant g_{\gamma+1, j}^{(i)}, \quad i = 1, 2, \ldots, m; \tag{7.20}$$

de plus, quelle que soit la valeur fixée de l'indice j, l'une au moins des relations (7.20) est une inégalité stricte.

Avant de décrire l'algorithme de résolution du problème (7.15)-(7.17), introduisons la notion de programme de base de ce problème.

Soit $X = (x_1, x_2, \ldots, x_n)$ un programme arbitraire du problème (7.15)-(7.17). Associons à toute composante x_j ($j = 1, 2, \ldots, n$) du programme X ne coïncidant avec aucune de ses valeurs critiques le vecteur

$$g'_j(x_j) = (g'_{1j}(x_j), g'_{2j}(x_j), \ldots, g'_{mj}(x_j))^T,$$

composé de dérivées $g'_{ij}(x_j)$ des fonctions $g_{ij}(x_j)$.

Si

$$d_{k_j j} < x_j < d_{k_j+1, j},$$

la formule (7.19) entraîne

$$g'_j(x_j) = (g_{k_j j}^{(1)}, g_{k_j j}^{(2)}, \ldots, g_{k_j j}^{(m)})^T.$$

Introduisons les composantes supplémentaires

$$x_{n+i} = -\sum_{j=1}^{n} g_{ij}(x_j), \quad i = 1, 2, \ldots, m,$$

qui complètent (7.16) de façon à les transformer en égalités. A la composante supplémentaire x_{n+i} du programme correspond le vecteur unité e_i de dimension m avec l'unité à la i-ème place. Rangeons en un ensemble E_X les indices $j \leqslant n$ des composantes x_j différentes des valeurs critiques du j-ième argument et les numéros $n + i$ des composantes supplémentaires positives. Le programme X est dit de base si les vecteurs associés aux composantes x_j du programme avec $j \in E_X$ sont linéairement indépendants. Si le nombre des éléments de l'ensemble E_X est m, le programme de base X est non dégénéré. Nous supposerons dans ce qui suit que chacun des programmes de base du problème (7.15)-(7.17) est non dégénéré. Les vecteurs correspondant aux composantes x_j du programme de base X pour $j \in E_X$ forment la base du programme X. A la différence des problèmes de programmation linéaire et des problèmes de programmation linéaire par morceaux examinés ci-dessus, les vecteurs qui composent la base d'un programme du problème II sont définis, en général, non seulement par le numéro de la composante x_j correspondante, mais encore par la valeur de cette composante.

La procédure de résolution du problème (7.15)-(7.17) fondée sur la méthode primale du simplexe se ramène au déplacement suivant les programmes de base et se compose d'un nombre fini d'itérations.

Voici le schéma général de cette procédure.

7.7. Pour abréger les notations, supposons que le programme de base courant X du problème (7.15)-(7.17) a pour base la matrice

$$G_X = (g_1'(x_1), \ldots, g_t'(x_t), e_{t+1}, \ldots, e_m).$$

Les composantes x_j du programme de base X non dégénéré vérifient les conditions

$$d_{k_j j} < x_j < d_{k_j+1, j}, \quad j = 1, 2. \ldots, t,$$
$$x_j = d_{k_j j}, \quad j = t+1, \ldots, n,$$
$$x_{n+i} = 0, \quad i = 1, 2, \ldots, t,$$
$$x_{n+i} > 0, \quad i = t+1, \ldots, m.$$

Avant de commencer l'itération, on suppose connue la matrice inverse G_X^{-1} de la matrice de la base G_X. A la première étape de l'itération, on teste l'optimalité du programme de base disponible. On définit dans ce but les composantes λ_i, $i = 1, 2, \ldots, m$, du vecteur Λ à partir du système d'équations suivant :

$$\Lambda G_X + C_X = 0,$$

où

$$C_X = (c_{k_1 1}, c_{k_2 2}, \ldots, c_{k_t t}, \underbrace{0, 0, \ldots, 0}_{m-t}).$$

On a

$$\Lambda = -C_X G_X^{-1}.$$

Cherchons ensuite les valeurs

$$\bar{\Delta}_j = \sum_{i=1}^{m} \lambda_i g_{k_j j}^{(i)} + c_{k_j+1, j},$$

$$\bar{\bar{\Delta}}_j = -\sum_{i=1}^{m} \lambda_i g_{k_j-1, j}^{(i)} - c_{k_j j}$$

$$\text{pour } j = t+1, t+2, \ldots, n,$$

où, comme d'habitude, $c_{-1j} = -\infty$, $c_{l_j+1, j} = \infty$.

Si

$$\bar{\Delta}_j \geqslant 0, \quad \bar{\bar{\Delta}}_j \geqslant 0, \quad \text{pour } j = t+1, \ldots, n, \tag{7.21}$$
$$\lambda_i \geqslant 0 \quad \text{pour } i = 1, 2, \ldots, t, \tag{7.22}$$

alors, d'après le critère d'optimalité donné au paragraphe 2.3, le programme X est solution du problème. Mais si certaines des rela-

tions (7.21), (7.22) ne sont pas respectées, on poursuit la procédure de résolution en passant à la deuxième étape de l'itération.

7.8. La deuxième étape de l'itération débute par le choix de la plus petite des grandeurs figurant dans les relations (7.21), (7.22). Désignons cette grandeur par Δ. Supposons qu'il s'agisse de

$$\Delta = \bar{\Delta}_r < 0, \quad r \geqslant t + 1.$$

Dans ce cas, le programme de base X subit une modification due à l'accroissement de sa r-ième composante. Plus précisément, on introduit le vecteur $X(\theta)$ aux composantes $x_j(\theta)$ données par la formule

$$x_j(\theta) = \begin{cases} x_j - x_{jr}\theta & \text{si} \quad j \in E_X, \\ x_j + \theta & \text{si} \quad j = r, \\ x_j \text{ dans les autres cas.} \end{cases} \tag{7.23}$$

On désigne ici par x_{jr} les coefficients de décomposition du vecteur $(g^{(1)}_{k_r r}, g^{(2)}_{k_r r}, \ldots, g^{(m)}_{k_r r})^T$ par rapport aux vecteurs de la base G_X. Le paramètre $\theta = \theta'$ est choisi le plus grand possible, mais tel qu'aucune des composantes $x_j(\theta)$, $j \in E_X$, $x_r(\theta)$ ne passe par sa valeur critique. Autrement dit, θ' est égal au plus petit des trois nombres

$$\min_{x_{jr}>0} \frac{x_j - d_{k_j j}}{x_{jr}}, \quad \min_{x_{jr}<0} \frac{x_j - d_{k_j+1,\, j}}{x_{jr}},$$

$$d_{k_r+1,\, r} - x_r.$$

Il faut retenir, de plus, que pour la composante supplémentaire $x_j > 0$,

$$d_{k_j j} = 0, \quad d_{k_j+1,\, j} = \infty.$$

Si $\theta' = \infty$, le problème étudié est irréalisable du fait que la fonction (7.15) n'est pas bornée sur l'ensemble (7.16)-(7.17).

Si par contre $\theta' < \infty$, nous passons à un nouveau programme de base $X' = X(\theta')$. La base du programme X' coïncide avec celle du programme X ($\theta' = d_{k_r+1,\, r} - x_r$) ou bien se forme à partir de ce dernier en substituant à l'un de ses vecteurs le vecteur

$$(g^{(1)}_{k_r r}, g^{(2)}_{k_r r}, \ldots, g^{(m)}_{k_r r})^T.$$

Il se peut que

$$\Delta = \lambda_r < 0.$$

Ce cas est analogue au cas examiné ci-dessus, à cette différence près que la modification du programme X est obtenue en augmentant la composante supplémentaire $x_{n+r} = 0$ associée au vecteur unité e_r.

Enfin, la dernière éventualité

$$\Delta = \bar{\bar{\Delta}}_r < 0, \qquad r \geqslant t + 1,$$

rend nécessaire la diminution de la composante $x_r = d_{k_r r}$ du programme disponible.

La transformation des composantes du programme X est définie par la formule

$$x_j(\theta) = \begin{cases} x_j + x_{jr}\theta, & j \in E_X, \\ x_j - \theta, & j = r, \\ x_j \text{ dans les autres cas.} \end{cases} \tag{7.24}$$

La valeur θ' du paramètre θ est posée égale au plus petit des nombres

$$\min_{x_{jr}>0} \frac{d_{k_j+1,j} - x_j}{x_{jr}}, \quad \min_{x_{jr}<0} \frac{d_{k_j j} - x_j}{x_{jr}},$$

$$x_r - d_{k_r-1,r}.$$

Avec $\theta' = \infty$, on établit que le problème n'a pas de solution. Mais si $\theta' < \infty$, on passe à un nouveau programme de base $X' = X(\theta')$. Dans tous les cas considérés, le passage à un nouveau programme de base X' conduit à diminuer la fonction économique de la quantité

$$F(X) - F(X') = -\Delta\theta'.$$

7.9. Comme le lecteur a dû le remarquer, la règle de calcul de la valeur θ' du paramètre θ dans l'algorithme exposé diffère de la règle analogue des algorithmes décrits précédemment.

Dans les algorithmes conçus pour l'analyse des problèmes de programmation linéaire par morceaux à conditions linéaires, la valeur du paramètre θ est définie par minimisation de la fonction

$$F(X(\theta))$$

par rapport à toutes les valeurs de θ telles que $X(\theta)$ est un programme du problème. Ce mode de choix de θ est défini par le déplacement à travers une série de valeurs critiques. Dans le cas de conditions linéaires, le passage par la valeur critique de l'une des variables ne change pas la base du programme de base, mais conduit seulement à modifier le critère du vecteur à ajouter. Si par contre les conditions du problème sont non linéaires, le passage par une valeur critique conduit à une modification de la base. C'est pourquoi l'application de la règle de sélection de la quantité θ, fondée sur la minimisation de la fonction $F(X(\theta))$, est associée dans ce cas

à plusieurs transformations de la matrice G_X. Cette circonstance nous a incités, lors de l'analyse du problème (7.15)-(7.17), à opter pour un mode plus simple, bien que moins efficace, de définition de la quantité θ. C'est ce mode qui a été décrit au paragraphe précédent.

Conformément au schéma général de l'algorithme exposé aux paragraphes 7.7 et 7.8, le passage du programme de base disponible au programme de base successif entraîne la modification de la matrice G_X^{-1} d'ordre m. Mais comme nous l'avons vu, la matrice de la base G_X compte $m - t$ vecteurs unités. Aussi le schéma de calcul peut être modifié de façon à conserver et à transformer en cours de résolution la matrice $\bar{G}_X^{-1}$ inverse de la sous-matrice $\bar{G}_X$ formée à l'aide des t premières lignes et des t premières colonnes de la matrice G_X. Nous laissons au lecteur le soin de le faire par lui-même en s'appuyant sur l'algorithme de résolution du problème (7.1)-(7.3) (cf. exercice 10).

7.10. Les schémas généraux des algorithmes de résolution des problèmes (7.1)-(7.3) et (7.15)-(7.17) exposés dans ce paragraphe servent de base à la composition des algorithmes pour la résolution d'autres problèmes de programmation linéaire par morceaux. Ainsi le schéma de calcul du problème III ne diffère que de peu de l'algorithme de résolution du problème (7.15)-(7.17). La différence ne porte seulement que sur les faits suivants. Premièrement, au lieu des vecteurs $g_j'(x_j)$ il faut considérer les vecteurs $(g_j'(x_j), A_j)^T$, dont les dernières composantes ne dépendent pas de x_j. Deuxièmement, pour les conditions linéaires du type égalités, la nécessité de variables additionnelles, naturellement, disparaît. En tirant parti des considérations émises aux paragraphes précédents, il est aisé aussi de tracer les contours généraux de l'algorithme de résolution du problème VII, dont les modèles examinés plus haut constituent tous des cas particuliers. Le passage de l'algorithme du problème (7.15)-(7.17) à cet algorithme plus complexe peut se faire par le procédé exposé aux paragraphes 7.1-7.4, où l'on construit sur la base du schéma de calcul du problème (3.1)-(3.3) un algorithme de résolution du problème (7.1)-(7.3). Nous laissons au lecteur le soin d'énoncer les principes généraux des algorithmes destinés à la résolution des problèmes IV, VI et VII (cf. exercice 12).

En conclusion de ce paragraphe, soulignons que les algorithmes décrits ici ne doivent être considérés que comme des schémas généraux de procédures de calcul établis pour l'exploration des problèmes de programmation linéaire par morceaux. L'application de ces algorithmes à chaque classe bien limitée concrète de problèmes de programmation linéaire par morceaux nécessite de détailler leurs procédures, afin de répondre le plus complètement possible aux particularités spécifiques d'un modèle donné.

Exercices

1. Montrer que la fonction

$$F(X) = \max_{1 \leqslant i \leqslant s} \{L_i(X)\},$$

où

$$L_i(X) = \sum_{j=1}^{n} a_{ij}x_j - b_i \quad i = 1, 2, \ldots, s,$$

est convexe vers le bas.

2. Justifier l'équivalence des problèmes (C) et (D) (cf. le paragraphe 1.4).

3. Montrer que les conditions (2.28)-(2.30) sont suffisantes pour rendre le programme du problème I optimal.

4. Justifier les conditions (2.33)-(2.35) comme suffisantes pour rendre le programme du problème II optimal.

5. Démontrer que le programme

$$Z^* = \{y^*_{01}, \ldots, y^*_{0s_0}, y^*_{11}, \ldots, y^*_{1s_1}, \ldots, y^*_{m_1 s_{m_1}}, x^*_1, x^*_2, \ldots, x^*_n\}$$

du problème IV est un programme optimal si et seulement si il est possible de composer des collections de nombres

$$\lambda_i \geqslant 0 \quad (i = 1, 2, \ldots, m_1), \quad \lambda_{ik} \ (k = 1, 2, \ldots, s_i;\ i = 0, 1, \ldots, m_1),$$
$$\lambda_i \ (i = m_1 + 1, \ldots, m),$$

telles que soient observées les conditions (2.47)-(2.52).

6. Déduire du critère d'optimalité du problème VII les critères d'optimalité des problèmes I à VI.

7. Donner la justification de la règle de choix du vecteur à éliminer de la base rendant impossible le cyclage dans les problèmes I dégénérés (cf. § 4).

8. Enoncer pour le problème I les principes de construction d'un algorithme de programmation linéaire par morceaux fondé sur la méthode duale du simplexe.

9. Minimiser la fonction linéaire par morceaux (6.5) sous les conditions (6.2) et (6.6).

10. Modifier l'algorithme exposé aux §§ 7.6-7.9 en utilisant l'existence de $m - t$ vecteurs unités dans la matrice de la base G_X du problème II.

11. Décrire le schéma de calcul de la résolution du problème III.

12. Enoncer les principes de résolution des problèmes IV, VI et VII.

13. Construire les schémas blocs de la résolution des problèmes I à VII.

14. Construire et décrire la méthode et l'algorithme de calcul du vecteur $X = (x_1, x_2, \ldots, x_n)$ sur lequel la fonction

$$\sum_{i=1}^{m} |L_i(X)|$$

est minimisée sous les conditions

$$\alpha_j \leqslant x_j \leqslant \beta_j, \qquad j = 1, 2, \ldots, n.$$

Ici

$$L_i(X) = \sum_{j=1}^{n} a_{ij}x_j - b_i, \quad i = 1, 2, \ldots$$

15. Enoncer le critère d'optimalité d'un programme du problème suivant de programmation linéaire par morceaux [32]:

$$z \to \min,$$

$$\sum_{j=1}^{n} |a_{ij}+b_{ij}x_j| - z \leqslant 0, \quad i=1, 2, \ldots, m_1,$$

$$\sum_{j=1}^{n} |a_{ij}+b_{ij}x_j| - b_i \leqslant 0, \quad i=m_1+1, \ldots, m,$$

$$\alpha_j \leqslant x_j \leqslant \beta_j, \qquad j = 1, 2, \ldots, n.$$

16. Construire l'algorithme de résolution du problème précédent.

17. Comparer le volume de calcul imposé par les deux approches suivantes de la résolution du problème (6.1)-(6.4). La première ramène le problème (6.1)-(6.4) au problème linéaire par morceaux équivalent (6.5), (6.2), (6.6). La deuxième considère le problème (6.1)-(6.4) comme un problème de programmation par blocs et utilise la méthode de décomposition (chapitre 4, § 2). De plus, les conditions des systèmes (6.3)-(6.4) forment des blocs isolés du problème, et les contraintes du système (6.2) constituent les conditions d'« enchaînement ».

18. Calculer le programme optimal du problème

$$|2x_1 - x_2 + 4| + |x_2 + 3x_3 - 2| + |x_3 - 2x_4 + 1| \to \min,$$

$$|3x_1 + x_2 - 1| + |2x_3 - 3x_4 + 2| - 4 \leqslant 0,$$

$$|x_1 + 2x_2 + 1| + |3x_3 + x_4 - 1| - 3 \leqslant 0,$$

$$|2x_1 + 3x_2 - 4| + |x_3 - 4x_4 + 3| - 5 \leqslant 0,$$

$$-1 \leqslant x_j \leqslant 1, \quad j = 1, 2, 3, 4.$$

BIBLIOGRAPHIE

1. ABADIE I. M., WILLIAMS A. C. Dual and parametric method in decomposition. « Recent Advances Math. Programm. », New York — San Francisco — Toronto — London, Mc. Graw-Hill Book Co., Inc., 1963, pp. 149-158.

2. AGHAR W. G., WALACE T. D. A sampling study of minimum absolute deviations estimators. Op. Res., 1963, 11, n° 5, pp. 747-758.

3. АМВРОСИЕНКО В. В. Ускорение сходимости метода Брауна решения матричных игр. « Экономика и математические методы » (AMVROSSIENKO V. V. Accélération de la convergence de la méthode de Brown pour la résolution des jeux matriciels. « Economie et méthodes mathématiques »), 1965, n° 4.

4. BALAS E. Un algorithme additif pour la résolution des programmes linéaires en variables bivalentes. CR Acad. Sci. Paris, 1964, 258, n° 15, pp. 3817-3820.

5. BELLMAN R. Dynamic Programming. Princeton Univ. Press, N.Y., 1957.

6. BELLMAN R. On a routing problem. Quart. Appl. Math., 1958, v. 16, n° 1.

7. BOWMAN E. H. The schedule-sequencing problem. Oper. Res., 1959, v. 7, n° 5, pp. 621-624.

8. BROWN G. W. Iterative solutions of games by fictitions play. Activity Analysis of Production and Allocation, ed. by T. C. Koopmans, Cowles Commission for Research in Economics Monograph n° 13, Wiley, New York, 1951, pp. 374-376.

9. BROWN G. W., VON NEUMANN J. Solution of games by differential equations. Contributions to the theory of games, v. I, 1950. Princeton, pp. 73-79.

10. БУЛАВСКИЙ В. А. Итеративный метод решения общей задачи линейного программирования. Сб. « Численные методы оптимального планирования ». Экономико-математическая серия, выпуск I. Изд-во Сиб. отд. АН СССР, Новосибирск (BOULAVSKI V. A. Méthode itérative de résolution du problème général de la programmation linéaire. Recueil « Méthodes numériques de planification optimale ». Série économie mathématique, fascicule I. Editions de la Section sibérienne de l'Académie des Sciences de l'U.R.S.S., Novossibirsk), 1962, pp. 35-64.

11. VAIDA S. Inequalities for stochastic linear programming. Bulletin of the International Statistical Institute, 1958, v. 36, pp. 357-363.

12. WAGNER H. M. An integer linear-programming model for machine scheduling. Nav. Res. Logist. Quart., 1959, v. 6, n° 2, pp. 131-140.

13. ВЕРХОВСКИЙ Б. С. О многоиндексной транспортной задаче с аксиальными суммами. ДАН СССР (VERKHOVSKI B. S. Sur le problème de transport à indices multiples et à sommes axiales. Comptes rendus de l'Académie des Sciences de l'U.R.S.S.), 1964, t. 156, n° 2, pp. 282-285.

14. WILLIAMS A. C. A treatment of transportation problems by decomposition. L. Soc. Ind. and Appl. Math., 1962, v. 10, n° 1, pp. 35-48.

15. WILLIAMS A. C. Marginal values in linear programming. J. Soc. Ind. and Appl. Math., 1963, v. 11, n° 1, pp. 82-94.

16. ВОЛКОНСКИЙ В. А. Оптимальное планирование в условиях большой размерности (итеративные методы и принцип декомпозиции) « Экономика и математические методы » (VOLKONSKI V. A. Planification optimale dans les conditions des problèmes de grande taille (méthodes itératives et principe de décomposition). « Economie et méthodes mathématiques »), 1965, n° 2.

17. WOLFE P., DANTZIG G. B. Linear programming in Markow chains. Oper. Res., 1962, v. 10, n° 5.

18. ГАЛЬПЕРИН М. В., КОРОТКЕВИЧ Г. П., МИНСКЕР И. Н., РЫБАСОВ В. И. К решению задач нелинейного программирования с одним и многими экстремумами на аналоговых вычислительных устройствах. « Техническая кибернетика » (GALPERINE M. V., KOROTKEVITCH G. P., MINSKER I. N., RYBASSOV V. I. Sur la résolution des problèmes de programmation non linéaire à un ou plusieurs extremums sur des calculateurs analogiques. Cybernétique technique), 1964, n° 4.

19. GASS S. I., SAATY T. L. Parametric Objective Function, Part II, Generalization. Journal of the Operation Research Society of America, 1955, v. 3.

20. GASS S. I., SAATY T. L. The computational algorithm for the parametric objective function. Nav. Res. Logist. Quart., v. 2, 1955.

21. ГОЛЬШТЕЙН Е. Г. Об одном видоизменении метода последовательного улучшения плана. « Экономика и математические методы » (GOLSTEIN E. G. Sur une modification de la méthode primale du simplexe. « Economie et méthodes mathématiques »), 1965, n° 1, pp. 83-86.

22. ГОЛЬШТЕЙН Е. Г. Об одном классе нелинейных экстремальных задач. ДАН СССР (GOLSTEIN E. G. Sur une classe de problèmes extrémaux non linéaires. Comptes rendus de l'Académie des Sciences de l'U.R.S.S.), 1960, t. 133, n° 3.

23. ГОЛЬШТЕЙН Е. Г. О возможности расширения применимости частных методов линейного программирования. Сб. « Планирование и экономико-математические методы », « Наука » (GOLSTEIN E. G. Sur l'extension éventuelle de l'application des méthodes particulières de programmation linéaire. Recueil « Planification et méthodes d'économie mathématique »). « Science », 1964, pp. 409-423.

24. ГОЛЬШТЕЙН Е. Г. Транспортная задача и ее обобщения. Сб. « Методы и алгоритмы решения транспортной задачи ». Госстатиздат (GOLSTEIN E. G. Le problème de transport et ses généralisations. Recueil « Méthodes et algorithmes de résolution du problème de transport »). Gosstatizdat, 1963.

25. ГОЛЬШТЕЙН Е. Г., ЮДИН Д. Б. Об одном классе задач планирования народного хозяйства. « Проблемы кибернетики » (GOLSTEIN E. G., YOUDINE D. B. Sur une classe de problèmes de planification de l'économie nationale. « Problèmes de cybernétique »), 1961, fascicule 5, pp. 165-182.

26. GOMORY R. E. All-integer programming algorithm. IBM, Res. Center Report RC-189, January, 29, 1960.

27. GOMORY R. E. An algorithm for integer solutions to linear programs. Princeton-IBM, Mathematics Research Project Technical Report, n° 1, November, 17, 1958.

28. GOMORY R. E. An algorithm for the mixed integer problem. RAND Corporation, P-1885, Santa-Monica, California, February, 22, 1960.

29. GOMORY R. E. Outline of an algorithm for integer solutions to linear programs. Bull. Amer. Math. Soc., 1958, 64, 5, pp. 275-278.

30. GOMORY R. E., HOFFMAN A. J. On the convergence of an integer-programming process. Nav. Res. Logist. quart. 1963, v. 10, n° 2, pp. 121-124.

31. HOFFMAN A. J., KRUSKAL J. B. Integral boundary points of convex polyhedra, dans [68].

32. ГНОЕНСКИЙ Л. С., МОВШОВИЧ С. М. О применении методов линейного программирования к одной задаче теории следящих систем. Изв. АН СССР. « Энергетика и автоматика » (GNOÉNSKI L. S., MOVCHOVITCH S. M. Sur l'application des méthodes de programmation linéaire à un problème de la théorie des systèmes asservis. Nouvelles de l'Académie des Sciences de l'U.R.S.S. « Energétique et automation »), 1962, n° 6, pp. 50-66.

33. ГУТЧИН И. Б., КУЗИЧЕВ А. С. Об оптимальном синтезе формальных нейронов. Изв. АН СССР. « Техническая кибернетика » (GOUTTCHI-

NE I. B., KOUZITCHEV A. S. Sur la synthèse optimale des neurones formels. Nouvelles de l'Académie des Sciences de l'U.R.S.S., « Cybernétique technique »), 1956, n° 2, pp. 47-57.

34. DANTZIG G. B. A proof of the equivalence of the programming and the game problem. Activity Analysis of Production and Allocation, ed. by Koopmans T. C., Cowles Commission Monograph, n° 13, New York, Wiley, 1951, pp. 330-335.

35. DANTZIG G. B. Linear programming and extensions (Univ. of Calif., Berkeley), 1963, Princeton Univ. Press, Princeton, New Jersey.

36. DANTZIG G. B. Note on solving linear programs in integers. Nav. Res. Logist. Quart. 1959, v. 6, n° 1, pp. 75-76.

37. DANTZIG G. B. On the significance of solving linear programming problems with some integer variables. Econometrica, 1960, v. 28, n° 1, pp. 30-44.

38. DANTZIG G. B., WOLFE P. Decomposition algorithm for linear programs. Econometrica, 1961, v. 29, n° 4, pp. 767-778.

39. DANTZIG G. B., WOLFE P. Decomposition principle for linear programs. Oper. Res., 1960, v. 8, 1, pp. 101-112.

40. DANTZIG G. B., MADANSKY A. On the solution of two-stage linear programming problems under unsertainty. Proc. Fourth Berkeley Symposium on Mathematical Statistics and Probability, 1961, n° 1.

41. DOIGE A., LAND A. H. An automatic method of solving discrete programming problems. Econometrica, 1960, v. 28, pp. 497-520.

42. ЗУХОВИЦКИЙ С. И. Об одной задаче кусочно-линейного программирования. Ж. « Вычислительная математика и математическая физика » (ZOUKHOVITSKI S. I. Sur un problème de programmation linéaire par morceaux. Journal d'analyse numérique et de physique mathématique), 1963, tome 3, n° 3, pp. 599-605.

43. IOSIFESCU M., THEODORESCU R. Sur la programmation linéaire. Comptes rendus hebdomadaires des séances de l'Académie des Sciences, 1963, v. 256, n° 253, pp. 4831-4833.

44. КАРАБЕГОВ В. И. Об одной параметрической задаче линейного программирования. Ж. « Вычислительная математика и математическая физика » (KARABEGOV V. I. Sur un problème paramétrique de programmation linéaire. Journal d'analyse numérique et de physique mathématique), 1963, tome 3, pp. 547-558.

45. KELLEY J. E. The « cutting plane » method for solving convexe programs. J. Soc. Ind. and Appl. Math., v. 8, n° 4, 1960, pp. 703-712.

46. LE GARFF A., MALGRANGE Y. Résolution des programmes linéaires à valeurs entières par une méthode booléienne « compacte ». Actes 3e conf. internat. rech. opérationnelle. Oslo, 1963, Paris — London, 1964, pp. 695-701.

47. LITTLE I., MURTY K., SWENEY D., KAREL C. An algorithm for traveling salesman problem. Oper. Res., 1963, v. 11, n° 6.

48. LUCE R. D., RAIFFA H. Games and Decisions. Wiley, 1957.

49. MADANSKY A. Inequalities for stochastic linear programming problems. Manag. Sci. 1960, v. 6, n° 2, pp. 197-204.

50. MAC KINSEY J. An Introduction to theory of games. Mc. Graw-Hill, N.Y., 1952.

51. MANNE A. S. Linear programming and sequential decisions. Manag. Sci., 1960, v. 6, n° 3.

52. MANNE A. S. On the job-shop sheduling problem. Oper. Res., 1960, v. 8, n° 2, pp. 219-223.

53. MARTIN G. T. An accelerated Euclidean algorithm for integer linear programming. Dans le receuil « Recent Advances in Math. Programm. », Mc-Graw Hill, New York, 1963, pp. 311-317.

54. МАШ В. А. Об одном способе выбора вводимой переменной в симплексном методе линейного программирования. Ж. « Вычислительная математика и математическая физика » (MACH V. A. Sur une méthode de sélection

de la variable à introduire dans la méthode du simplexe de programmation linéaire. Journal d'analyse numérique et de physique mathématique), 1964, tome 4, n° 2.

55. MILLER C. E., TUCKER A. W., ZEMLIN R. A. Integer programming formulation of travelling salesman problems. J. Assoc. Comput. Machinery, 1960, v. 7, n° 4, pp. 326-329.

56. MILLS H. D. Marginal values of matrix games and linear programms dans [68].

57. MINTY G. A comment on the shortest-route problem. Oper. Res., 1957, v. 5, p. 724.

58. МУДРОВ В. И. Замечание к статье С. И. Зуховицкого « Об одной задаче кусочно-линейного программирования ». Ж. « Вычислительная математика и математическая физика » (MOUDROV V. I. Note sur l'article de S. I. Zoukhovitski « Sur un problème de programmation linéaire par morceaux ». Journal d'analyse numérique et de physique mathématique, 1964, tome 4, n° 5, pp. 968-969.

59. NEUMANN J., VON. A numerical method to determine optimum strategy. Nav. Res. Logist. Quart. 1954, v. 1, n° 2, pp. 109-115.

60. PYNE I. B. Linear programming on an analog computer. Trans. Paper AIEE, Princeton New Jersey, Princ. Univ., 1956, pap. n° 56, p. 147.

61. PIETSZYKOWSKI T. An iteration method of linear programming. Zaklad. Aparatow Matematycznich PAN, Prace A, n° 5, 1960, Warczawa.

62. PIETSZYKOWSKI T. Application of the steepest ascent method to linear programming. Zaklad Aparatow Matematycznich PAN, Prace A, n° 11, 1961, Warczawa.

63. POLLAC M., WIEBENSON W. Solutions of the shortest-route problem — a review. Oper. Res., 1960, v. 8, n° 2, pp. 224-230.

64. RADNER R. The application of linear programming to team decision problems. Manag. Sci., 1959, v. 5, n° 2, pp. 143-150.

65. REITER. Surrogates for uncertainty decision problems: minimal information for decision making. Econometrica, 1957, v. 25, n° 2.

66. ROBINSON J. An iterative method of solving a game. Ann. Math. 54 (1951), pp. 296-301.

67. SAATY T. L. Coefficient perturbation of a constrained extremum. Oper. Res., 1959, v. 7, n° 3.

68. KUHN H. W., TUCKER A. W. (editors). Linear inequalities and related systems. Annals of Mathematics Study n° 38, Princeton University Press, Princeton (New Jersey), 1956.

69. UZAWA H. Voir le recueil: Arrow K. J., Hurwicz L. et Uzawa H., Studies in linear and non-linear programming, Stanford University Press, Stanford (California), 1958.

70. FERGUSON R. O., DANTZIG G. B. The allocation of airkraft to routes.— An example of linear programming under uncertainty. Manag. Sci., 1956, v. 3, n° 1, pp. 45-73.

71. ФИНКЕЛЬШТЕЙН Ю. Ю. Алгоритм для решения задач целочисленного программирования с булевыми переменными. « Экономика и математические методы » (FINKELSTEIN Y. Y. Un algorithme de résolution des problèmes de programmation en nombres entiers à variables booléiennes. Economie et méthodes mathématiques), 1965, tome 1, n° 5.

72. ФИНКЕЛЬШТЕЙН Ю. Ю. О решении задач дискретного программирования специального вида. « Экономика и математические методы » (FINKELSTEIN Y. Y. Sur la résolution des problèmes de programmation discrète d'un type spécial. Economie et méthodes mathématiques), 1965, tome 1, n° 2.

73. FAURE R., MALGRANGE Y. Une méthode booléienne pour la résolution des programmes linéaires en nombres entiers. Gestion, 1963, v. 6, pp. 250-260.

74. FORD L. R., FULKERSON D. R. A simple algorithm for finding maximal network flows and an application to the Hitchcock problem. Nav. Res. Logist. Quart., 1957, v. 4, n° 1, pp. 45-54.

75. FORD L. R., FULKERSON D. R. Maximal flow through a network. Canadian J. Math., 1956, v. 8, pp. 399-404.

76. HELLER I., TOMPKINS C. B. An extension of a theorem of Dantzig's, article dans le recueil [68].

77. CHARNES A., COOPER W. W. Deterministic equivalents for optimizing and satisfying under chance constraints. Oper. Res., 1963, v. 11, n° 1, pp. 18-39.

78. CHARNES A., COOPER W. W., SYMONDS G. H. Cost horizon and certainty equivalent. Manag. Sci., 1958, v. 4, n° 3, pp. 235-263.

79. ЧЕРЕНИН В. П. Решение некоторых комбинаторных задач оптимального планирования методом последовательных расчетов. « Научно-методические материалы экономико-математического семинара ». ЛЭММ АН СССР, ВЦ АН СССР, 1962 (TCHÉRÉNINE V. P. Résolution de certains problèmes combinatoires de planification optimale par la méthode de calculs successifs. Matériaux scientifiques et méthodologiques du séminaire d'économie mathématique. Académie des Sciences de l'U.R.S.S. Centre d'analyse numérique de l'Académie des Sciences de l'U.R.S.S.), 1962, fascicule 2.

80. ЧЕРЕНИН В. П., ХАЧЕТУРОВ В. Р. Решение методом последовательных расчетов одного класса задач о размещении производства. Сб. « Экономико-математические методы », « Наука » (TCHÉRÉNINE V. P., KHATCHATOUROV V. R. Résolution par la méthode de calculs successifs d'une classe de problèmes de distribution de la production. Recueil « Méthode d'économie mathématique », « Science »), 1965, fascicule 2, pp. 279-289.

81. SHAPIRO H. N. Note on a computation method in the theory of games. Comm. Pure and Appl. Math., 11, n° 4 (1958), pp. 588-593.

82. SHAPLEY L. S., SNOW R. N. Basic solutions of discrete games. Contributions of the Theory of Games, v. 1, Annals of Math. Studies, Princeton, 1950, n° 24, pp. 27-35.

83. SHIMBELL. Structure in communication nets. Proceedings of the symposium on information networks, Polytechnic Institute of Brooklyn, April. 1954, pp. 12-14.

84. ARROW K. J., HURWICZ L. Gradient method for concave programming, I: local results (p. 117); dans le recueil: Studies in linear and non-linear programming, Stanford University Press, Stanford (California), 1958.

85. ARROW K. J., SULOW R. M. Gradient method for constrained maxima, with meakened assumptions; dans le recueil: Studies in linear and non-linear programming, Stanford University Press, Stanford (California), 1958.

86. ГОЛЬШТЕЙН Е. Г., ЮДИН Д. Б. Задачи и методы линейного программирования. Изд-во « Советское радио », 1961 (GOLSTEIN E. G., YOUDINE D. B. Problèmes et méthodes de programmation linéaire. « Radio soviétique »), 1961.

87. ГОЛЬШТЕЙН Е. Г., ЮДИН Д. Б. Задачи и методы линейного программирования. Издание 2-ое, переработанное и дополненное. Изд-во « Советское радио » (GOLSTEIN E. G., YOUDINE D. B. Problèmes et méthodes de programmation linéaire. 2e éd. revue et corrigée. « Radio soviétique »), 1964.

88. ГОЛЬШТЕЙН Е. Г., ЮДИН Д. Б. Линейное программирование (Теория и конечные методы). Физматгиз (GOLSTEIN E. G., YOUDINE D. B. Programmation linéaire (théorie et méthodes finies). Fizmatguiz), 1963.

89. ГОЛЬШТЕЙН Е. Г., ЮДИН Д. Б. Об одном методе количественного анализа упрощенных экономических моделей. Сб. статей « Применение математики в экономических исследованиях ». Соцэкгиз (GOLSTEIN E. G., YOUDINE D. B. Sur une méthode d'analyse quantitative des modèles économiques simplifiés. Recueil d'articles « Application des mathématiques dans les recherches économiques ». Sotsekguiz), 1962, tome II, pp. 136-199.

INDEX ALPHABÉTIQUE

Algorithme, 385
— de Gomory, 398
— de programmation en nombres entiers, 385, 396
— de programmation linéaire par morceaux, 507
— de programmation partiellement en nombres entiers, 407
— de résolution d'un problème, 211, 215
Algorithmes de programmation linéaire par morceaux, 536
Amvrossienko, V. V., 128, 134
Analogue dual, 306
Applications de la programmation paramétrique, 196
Arborescence, 91

Base du programme de base, 495
Base fondamentale finie, 180
Base optimale, 152
Bellman, R., 26
Bokova, V. V., 9
Boulavski, V. A., 134
Brown, G. W., 110, 318

Capacité de la coupe, 33
Cas général, 178
Cas particuliers et modifications, 257
Charnes, A., 441
Choix de l'approximation initiale, 323
— d'un système d'armement, 367
Communication X-insaturée, 34
Communications de base, 68
— d'un réseau, 11
— principales d'un programme, 51
Condition de bilan, 20
Conditions d'intégralité des ensembles convexes, 379
— d'optimalité du programme, 313
Contraintes bilatérales, 49
— induites, 452
Convergence, 123
Cooper, W. W., 441
Coupe du réseau de transport, 33
— minimale, 39
Couple admissible, 439
Coups aléatoires, 90
— personnels, 90
Critère d'optimalité, 71, 482
— — du programme d'un problème, 49
Cyclage, 397, 503, 525

Dantzig, G. B., 89, 257, 366, 385, 420, 452
Définition d'un programme, 432
Dégénérescence, 503, 525
Deuxième algorithme, 242, 495
Deuxième interprétation géométrique, 164
Deuxième théorème de dualité, 105, 232
Direction possible, 305, 313, 321

E-modèle, 436, 448
Ensemble convexe, 297
— des états de la nature, 433
— d'optimalité, 152, 166
— d'optimalité de la base, 180
Entrée, 31
Equivalence stratégique, 97
Erreur relative de détermination de la valeur du jeu après la s-ième partie, 113
Etat de la nature, 433
Evaluation des solutions, 463
EV-modèles, 436
Exploration d'un problème général à un paramètre de la programmation linéaire, 9
Exploration du Z-problème, 300

Facteur de qualité de la résolution d'un problème, 435
Flot algébrique, 20
— arithmétique, 19
— du réseau, 18
— maximal, 31, 33, 39, 42, 78
Fonction de fabrication et de consommation, 17
— de Lagrange, 220
— de paiement, 94
— des capacités, 20
— des capacités des communications, 17

BeatriceGloria_personal library

Fonction des coûts de transport, 21
— linéaire par morceaux, 472
— — — — à convexité dirigée vers le bas et avec une seule brisure, 46
Ford, L. R., 32
Forme développée de jeu, 91
— normale de jeu, 91
Formulation duale, 296
— en termes de jeu, 437
— — — de réseaux des problèmes de transport, 48
— matricielle des problèmes de transport, 48
— serrée d'un problème, 432, 439
Formulations d'un problème de transport en termes de réseau, 52
— matricielles des problèmes de transport, 51
Fulkerson, D. R., 32

Gass, S. I., 148
Gomory, R. E., 365, 385, 386, 420
Graphe, 16

Hoffman, A. I., 420

Itinéraire, 14
— fermé, 15
— optimal, 22
— orienté, 15
— *s*-optimal, 31
Itinéraires fermés, 71
— généralisés, 31, 72
— *s*-optimaux, 27

Jeu à somme nulle, 437
— avec information complète, 90
— avec information incomplète, 90
— matriciel, 104
— rectangulaire matriciel, 91
— symétrique, 101
Jeux matriciels, 96

Kelley, J. E., 456

Madansky, A., 452
Matrice de gain, 91
— de paiement, 91
Maximin, 95
Méthode de Bellman-Shimbell, 30, 56, 72, 88
— de Brown, 128, 134, 318
— de décomposition, 238, 257, 293, 306, 336, 345, 352, 359, 456
— — — appliquée à la matrice diagonale par blocs, 265
— — — appliquée au problème de transport, 339
— — — de Dantzig et Wolfe, 296, 303, 338
— de Minty, 41, 88
— de Neumann, 121
— de programmation dynamique, 425
— — — en nombres entiers, 391
— — — paramétrique, 320, 437
— de Pyne, 144
— de résolution, 324
— de Thomas Pietszykowski, 134, 139, 140
— des encadrements, 338
— des gradients, 145
— des itinéraires fermés, 71
— des potentiels, 11, 61, 63, 218
— duale du simplexe, 172, 322, 394
— du jeu fictif, 110, 118
— du simplexe, 118, 125
— hongroise, 11, 76, 84
— itérative, 121
— — de Brown, 112
— PERT, 22, 31
— primale-duale, 323
— primale du simplexe, 159, 218, 238, 253, 306, 495, 546
Méthodes combinatoires de résolution des problèmes discrets, 427
— de programmation en nombres entiers, 417
— — — par blocs fondées sur la minimisation de la fonction $f(\Lambda)$, 310, 327
— de résolution des jeux, 107
— finies, 7, 338, 472
— itératives, 8, 134
— primales du simplexe, 387
Minimax, 94
Minty, G. Y., 23
Mise en ordre lexicographique, 392
M-méthode, 208
Mm-modèle, 436
MmE-modèles, 436
Modèle de planification du volume de production, 480
Modèles de transport, 11

Modèles probabilistes dynamiques, 465
Modifications de la méthode de décomposition, 291

Neumann, J. von, 121

P-modèle, 436
Point critique, 163
— de consommation, 17
— de fabrication, 17
— de transit, 17
— « le plus bas », 300
— selle, 92, 94, 220
Points du réseau de transport, 11
— extrémaux de l'itinéraire, 14
— voisins, 62
Polyèdre convexe, 445, 484
— des conditions, 398
Potentiels des points du réseau, 50
Premier théorème de dualité, 97
Premier théorème du problème du flot maximal, 39
Problème à deux étapes, 458
— — — — à vecteur des contraintes aléatoire, 464
— à trois indices à sommes axiales, 355
— combinatoire, 369
— d'affectation, 369
— de distribution, 347, 367, 373, 428
— — — déterministe, 460
— — — en nombres entiers, 425
— de la coupe optimale des matériaux, 367
— de l'itinéraire le moins cher, 22
— de maximisation de la forme linéaire vectorielle, 392
— de minimisation de la fonction, 300,
— de planification à long terme, 448
— de programmation homographique, 208, 310
— — — linéaire, 196, 206, 318
— — — — en nombres entiers, 379
— — — — paramétrique, 166
— de transport, 52, 56
— — — sur le réseau, 68
— de type combinatoire, 369
— d'un chemin le plus court, 22
— d'un flot maximal, 31
— du sac à dos, 368, 423
— du voyageur de commerce, 370
Problème en nombres entiers, 399
— $\Gamma(M, \Lambda')$, 313
— $\widetilde{\Gamma}(M, \Lambda')$, 311
— non dégénéré, 495
— non linéaire à fonction économique discontinue, 377
— paramétrique, 207
— partiellement en nombres entiers, 411
Problèmes à contraintes probabilistes, 434, 441
— à deux étapes, 450
— à une étape, 439
— convexes, 373
— décomposés en blocs, 294
— de planification, 477
— de programmation linéaire, 99, 239, 245
— — — — en nombres entiers, 366
— — — stochastique, 481
— de transport, 11, 339
— — — à indices multiples, 354
— $\Gamma(M, \Lambda')$ et $\widetilde{\Gamma}(M, \Lambda')$, 328
— linéaires, 472
— — par morceaux, 493
Programme de base, 50, 68, 343, 495, 537, 538, 545
— — — du *Z*-problème, 249
— de transport, 20
— d'un problème, 432, 478
— non dégénéré, 61, 495
— optimal, 478, 517
— — cherché, 303
— — du problème en nombres entiers, 422
— permanent, 439
— convexe, 8, 445, 456
Programmation dynamique, 26
— en nombres entiers, 8, 420
— linéaire, 7, 96
— — en nombres entiers, 365
— — par morceaux, 8, 471
— — partiellement en nombres entiers, 366
— markovienne, 439, 465
— optimale, 8
— paramétrique, 9, 148, 149, 164, 171, 268, 308
— par blocs, 8, 238, 309, 322, 335
— stochastique, 8, 450, 465, 480
Pseudobase, 166
Pyne, I. B., 144

Recherche de la solution du jeu, 96
Répartition de l'équipement dans un avion, 367

Représentations matricielles, 11
Réseau connexe, 16
— de transport, 11, 16
Résolution par la méthode de décomposition, 343
Robinson, J., 110, 111, 318

Saaty, T. L., 148
Schéma de calcul, 250
— d'« information par habitude », 111
— général de résolution d'un *Z*-problème, 242
Segment de linéarité de la fonction, 75
Sensibilité des solutions des problèmes de programmation linéaire à la variation des données, 218
Shimbell, 26
Situation d'équilibre, 92
Situations comportant un risque, 480
— indéterminées, 480
Solution du jeu, 96
Solution en une seule étape, 432
Sous-réseau du réseau de transport, 16
— d'un réseau de transport maximal, 16
Stratégie d'un jeu, 91
— essentielle, 104
— inessentielle, 104
— mixte, 93
— optimale, 93
— — d'un joueur, 94
— pure, 91

Tableau auxiliaire, 511
— principal, 510, 526
Théorème de Robinson, 112
— de Shapley-Snow, 106
— fondamental de la théorie des jeux, 96
Théorie des horaires, 372
— des jeux, 7, 89, 90

Uzawa, H., 134
U-modèles, 436

Valeur du coût de transport, 22
— du flot, 18
— du jeu, 92, 94
Valeurs critiques de l'argument d'une fonction, 494
Variable d'écart, 64
Variables additionnelles, 391
— principales, 391
Vecteur des contraintes, 494
Vecteurs des conditions, 494
V-modèles, 436, 446
Volkonski, V. A., 128

X-problème, 240, 340

Z-problème, 241, 251, 343
Z-problème tronqué, 293